UNIVERSITÀ DI PISA

Department of Computer Science

Data and Models for Understanding and Generating Stylistic Variation in Language.

Author:
Lorenzo De Mattei

Supervisors:
Felice Dell'Orletta,
Malvina Nissim,
Dino Pedreschi

Abstract

Stylistic choices are a fundamental aspect of verbal communication both from the language understanding and from the language production points of view. The goal of this thesis is twofold. From one side the focus is on the study of Natural Language Processing techniques that are able to capture stylistic variations in texts. On the other side, the thesis investigates how those techniques can be applied to assess the capability of the Natural Language Generation System to reproduce such stylistic variations in generated texts.

To my family members Stefania, Maurizio, Valeria, Maria and the latest arrivals Anna Bianca and Giorgio.

Acknowledgements

I thank my supervisors Felice and Malvina for their precious teachings. They guided me to tackle scientific problems with great enthusiasm while giving me the freedom to express myself. Their contribution to my personal growth is invaluable. I also thanks my supervisor Dino Pedreschi and the internal committee members Davide Bacciu and Anna Monreale for their supervision and precious comments and feedback. I thank the external reviewer Paolo Rosso and Amy Isard for having provided extremely valuable feedback and ideas to improve my thesis. I thank all the co-authors, to name a few Michele, Dominique, Giulia, Marco, Albert, Huiyuan, who bring their amazing contributions to the work reported in this thesis. I thank my colleagues and office buddies, to name a few Said, Chiara, Alessio for their support. I thank the former director of the PhD course Paolo Ferragina and the current director Antonio Brogi who have been always available for support. I thank the Computer Science department of University of Pisa and the institute ILC-CNR Antonio Zampolli for having provided an amazing environment and all the tools I needed for my PhD project. I thank the Computational Linguistics group of the University of Groningen who adopted me during my period abroad. I thank my partner Andrea who acts like a brother to me. I thank my betrothed Maria, without her support I would not reached the goal of completing this thesis. I thank my sister Valeria who grew up with me and she is always available to help me. Finally, I thank my parents Stefania and Maurizio who have provided me with an extraordinary education since I was a child, without which I would never have achieved these goals, and I also thank them for their financial support.

Contents

Chapter 1

Introduction

1.1 Motivations

Most of the words we read on the Internet were written by humans, but that could soon change. Natural language processing (NLP), that *"is the field of designing methods and algorithms that take as input or produce as output unstructured, natural language data"* [115], during the last decade has witnessed an unprecedented leap in technological advancement, with much of the progress enabled by Artificial Neural Networks (ANNs). ANNs is a class of machine learning model that was inspired by the brain's computation mechanism, which consists of computation units called neurons. More specifically the progress in NLP are lead by Deep Learning: a term we refer to identify a part of the broader family of ANNs. The name Deep Learning stems from the fact that many layers of neurons are chained together. The reason why Deep Learning had such a huge impact in NLP is that while all of the machine learning can be characterized as learning to make predictions based on past observations, deep learning approaches work by learning not only to predict but also to correctly represent the data, such that it is suitable for prediction. Indeed in NLP data representation is the most complex aspect. This topic is widely discussed in Chapter 2.

Natural Language Generation (NLG) is a subfield of NLP *"that is concerned with the construction of computer systems that can produce understandable texts"* [252]. As for NLP the raising of Deep Learning boosted NLG research and sensational results have been achieved [35] [176]. For example, the language model GPT-3 [35] produced an article that has been published in the Guardian [120]. These models are so powerful that some concerns about the potential usage for malicious purposes has been raised by the research community: for example the research lab OpenAI originally announced the previous version of GPT-3, GPT-2, in February 2019, but withheld the full model out of fear it would be used to spread fake news, spam, and disinformation. Initially, they released smaller, less complex versions of GPT-2 and studied their reception. Others also replicated the work. In a blog post, OpenAI later said it is seen "no strong evidence of misuse" and has released the model in full [298].

The results obtained thanks to Deep Learning techniques are exciting but after all the ultimate goal of NLG systems is to generate texts that are valuable to people and people's language "is remarkably varied. There is variation across speakers, that is, reflections of different ways that people speak in different regions or social groups, but also variation within the speech of a single speaker. No one speaks the same way all the time, and people constantly

exploit variation within the languages they speak for a wide variety of purposes." [302]. The ability to understand and produce variation, sometimes called communicative competence, is strongly related to social aspects and is fundamental for efficient verbal communication. Indeed the discipline that is concerned with the descriptive study of linguistic variation is called sociolinguistics. In this thesis, we want to investigate computational methodology to understand and produce linguistic variation in texts.

1.2 Language Variation

Sociolinguistics is a wide field of study: during years a lot of theoretical research has been conducted on this topic. An extended review of the literature of sociolinguistics is out of the scope of this thesis, however in this section are briefly reported a few basic concepts that are useful to contextualize this research project.

[167] defines variants as different linguistics artefacts "identical in reference or truth value, but opposed in their social and/or stylistic significance". Specifically, we refer to the variable as a set of alternative methods to refer to the same thing, and to variant to each possible concrete realization of a variable. Variants have impacts at different linguistic levels: phonological, morphological, lexical and syntactical. Sociolinguists have identified at least 5 different axes of linguistic variation:

- diachronic variation: the variation through time;
- diatopic variation: the variation through different regions;
- diastratic variation: the variation through different social classes;
- diaphasic variation: the variation through different situations;
- diamesic variation: the variation through different media. [27]

These axes interact with each other: a concrete linguistic production will ever have a collocation along all of them. Moreover, variations in one of these axes might impact each other, for example, a variant originally clearly marked as diatopic might assume diastractic or diaphasic nuances [27].

From a theoretical point of view is not always straightforward to clearly define a set of variants in respect to a variable, indeed the boundary between semantic and social aspects might be not clearly defined.

In the scope of this thesis, we are specifically interested in stylistic variation. Following [105], we use the notion of linguistic style broadly, "operationalising it in the terms most relevant to the problem at hand. 'Style' is usually understood to refer to features of lexis, grammar and semantics that collectively contribute to the identifiability of an instance of language use as pertaining to a specific author, or to a specific situation". Despite that [167] specifies that variants do not involve semantic changes, considering the fact that there are no clear boundaries between semantic and variational changes, in this project we consider also phenomena that might involve semantic shifts but that reflects 'stylistic' variations,

in its broader notion. Indeed more in general the main goal of this thesis is to investigate computational methodology that embeds the 'how' of language, while the 'what' is considered only collaterally.

1.3 Understanding Variation

The first objective of this thesis is to contribute to computational methodologies to understand variation in texts. These methodologies are well studied by a subfield of NLP: Natural Language Understanding (NLU). NLU deals with machine reading comprehension and is not only focused on variational aspects. However, most of the modern NLU methodologies applies both to variational and semantic aspects modelling.

NLU scientific literature is wide, but a common trend is to reduce a wide set of tasks to sequence classification (or regression) problems. The modern common pattern is to take advantage of them by exploiting machine learning techniques, what varies between tasks are the target categories (or score in the case of regression) that model a vast amount of different aspects such as topic, author profiling, sentiment and many more. The classical approach to tackle those tasks is to manually engineer feature extraction techniques from texts and to use those features as input to the machine learning models. The machine learning techniques used are varied but a single technique can be used for a wide set of NLU tasks. What typically varies between each task is the features extraction procedure: the information needed to solve different tasks might be not the same. With the advent of Deep Learning the models themselves learns the features they need, so that the methodologies to solve different tasks are converging.

In this thesis, we experiment with those machine learning techniques on several variation related tasks with the final goal of understanding their potentiality and limitations so that we can exploit them consciously to model variation in automatically generated texts.

1.4 Generating Variation

The results obtained by modern NLG systems are encouraging although their evaluation is marked by a great deal of variety and it is difficult to compare systems performances directly [105]. A vast amount of evaluation techniques that consider different aspects of NLG systems has been proposed. Those techniques involve both human and automatic evaluation and consider both objective and subjective aspects.

The ultimate goal of NLG systems is to generate text that is valuable to people. For this reason, human evaluations are typically viewed as the most important form of evaluation for NLG systems [45]. At the same time, human evaluation also presents its challenges and there have been calls for the development of new, more reliable metrics [226]. Beyond the costs associated with using humans in the loop during development, it also appears that certain linguistic judgment tasks related to variation are hard for humans to perform reliably. For instance, human judges show relatively low agreement in the presence of syntactic variation [41]. By the same token, [85] observes at best moderate correlations between human raters

on stylistic dimensions such as politeness, colloquialism and naturalness. In profiling, where evaluation can be performed against discrete gold labels, several studies found that humans are definitely not better than machines in identifying the gender of a writer [161, 100, 118]. Similarly, humans failed to outperform automatic systems in recognising the native language of non-English speakers writing in English [200]. [14] also found that seven out of ten subjects, including professional translators, performed worse than a simple automatic classifier at the task of telling apart original from translated texts.

Intuitively if we are able to develop tools that are able to understand stylistic variations, we can then apply them to evaluate and improve the performances of NLG systems to produce such stylistic variations.

Exploiting NLU to assess the goodness of generated texts in connection to a broad definition of style-aware generation has been used in several previous works [138, 292, 241, 145, 178, e.g.]. Most of these works tend to focus on aspects, which are usually most associated with a lexical problem (only a small part of *style*). Indeed, the problem of style transfer is usually addressed within the Variational Autoencoder framework and/or through lexical substitution. The lexical substitution was also the key element of a system developed for obfuscating gender-related stylistics aspects in social media texts [247], where a classification-based evaluation was used. In addition, [178] compared the automatic classification-based evaluation with human evaluation. They find a high correlation between human and automatic evaluation in two out of their three data-sets, showing the validity of the automatic approach. However, the tasks considered, though subjective, are not too hard for humans. [246] also exploited human and automatic classification as benchmarks for a machine translation system that translates formal texts into informal texts and vice-versa. Also in this case, usually text register is something that humans are quite able to grasp.

In this thesis, we study approaches that differ from the previous works in at least two respects. One is that we want to evaluate the capabilities of an NLG system to learn (different) stylistics aspects, rather than evaluating the capabilities of style transfer systems mostly based on lexical substitution. The other is that the stylistic aspects that we attempt to model are not always easily identified by human annotators. Therefore, relying on human-based evaluation in a real setting is not always an option, and even the classification-based method cannot be easily validated against human judgement for some of the tasks we took into consideration.

1.5 Thesis Structure

In this section is reported briefly how the thesis is organized. Figure 1.1 summarizes the works done in the scope of this thesis, organized under two main axes: (i) NLU vs NLG and Objective Aspects vs Subjective aspects.

In Chapter 2 is reported a brief review of the NLP techniques relevant for the scope of this thesis. In Chapter 3 is reported a brief review of NLG, with a special focus on the evaluation of NLG systems.

One of the aspects we focused on is linguistic complexity and specifically perceived language complexity, this topic is reported in Chapter 4. In Chapter 5 is reported the research

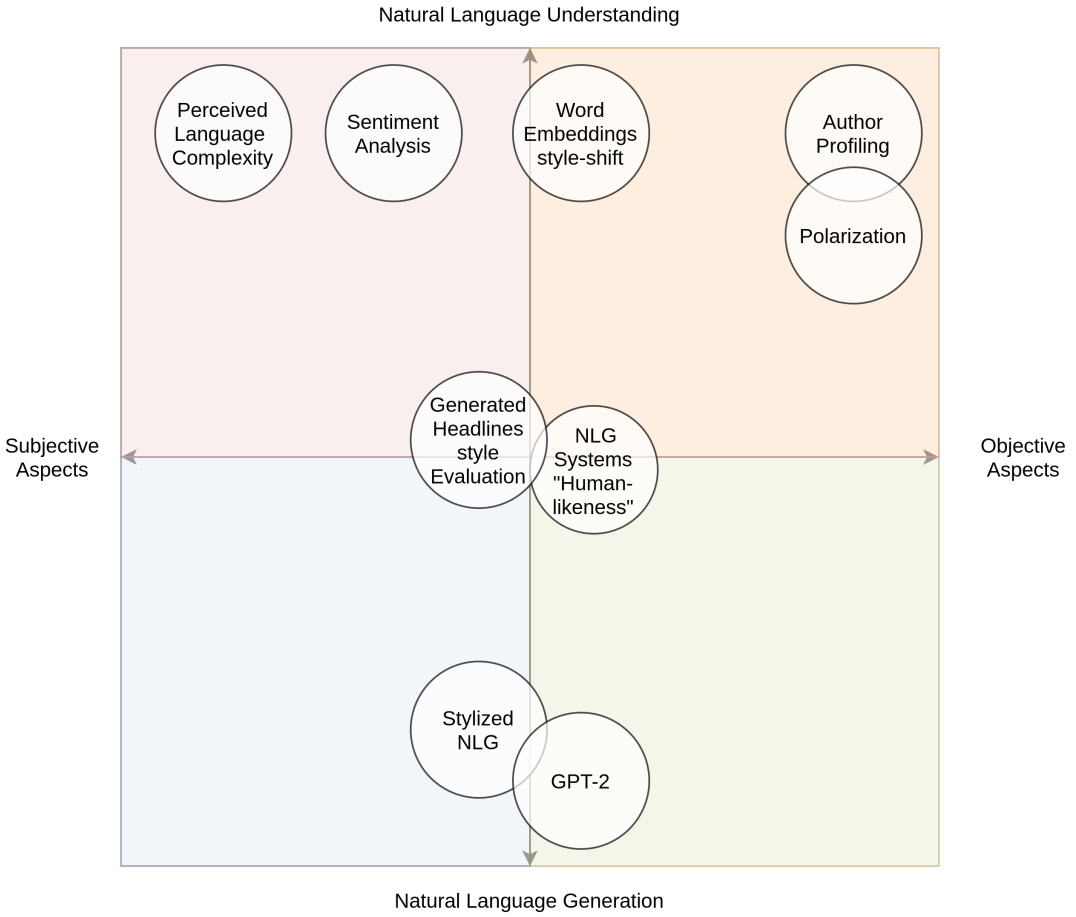


FIGURE 1.1: Summary of the topic explored in this thesis

conducted about text classification with a special focus on Multi-Task Learning (MTL). The discussed techniques have been successfully applied in several tasks such as sentiment analysis, irony detection, user profiling, hate speech detection establishing the state of the art on Italian for all of them. In Chapter 6 we will analyze stylistic aspects of polarization of news headlines, we will introduce a few NLG systems for polarized news headline generation, and we will analyze how human evaluation can be extended exploiting NLU techniques. In Chapter 7 we will describe GePpeTto: the first generative language model for Italian built using the GPT-2 architecture [243]. In the same chapter, we will deep dive into how humans perceive automatically generated texts and how NLU techniques can be exploited to assess and improve the "human-likeness" of text generated automatically. The Chapter 8 is dedicated to the conclusion and future works.

1.6 Main contributions

In the scope of this thesis, we successfully modelled several different variational aspects through NLU techniques on human-produced texts, establishing the state of the art for each task we took into consideration. More importantly from these studies, we derived a vast amount of hints about the capabilities and the limitations of the techniques we exploited.

Also, we produced several NLG systems exploiting different techniques for different goals and we used both human and automatic evaluation to assess the models' capabilities. By doing it we provided the NLP community with a lot of novel results about the reliability of NLU techniques for NLG systems evaluation and improvement.

All the works done for this thesis have been done for the Italian language at least (the language complexity analysis has been done also for English) for two main reasons:

- we consider fundamental for the Italian NLP community to have available as many NLP resources and models as possible, and by doing those work for Italian we produced data and models (all of them are publicly available),
- Italian is a low-resource language, then the work we have done is replicable for other languages. Research results on English the highest resource language, are not always replicable for low-resource languages.

The research we conducted has been done on short texts such as sentences, news headlines or social media posts. In this way, we have been able to produce results in low-resources settings.

1.7 Impact Statement

In the previous section are reported the main contributions we provided. These contributions can be clearly beneficial to society: for example the style modelling techniques can be exploited to provide services tailored to match specific needs of different kind of people. Also we provided several resources and models for Italian language, enabling the development of services that could help the digitalization process of our country. This said, some of the model we built and released publicly in the scope of this thesis could unfortunately be used maliciously. It is the case for example of the author profiling systems we developed that can be used to mine personal information of the analyzed texts without their authors' authorization, or of the generative models that might be used to generate texts for malicious scopes (eg. fake news generation). Also the neural model we largely adopted lack of explainability and might embed biases contained in the training data, therefore even if those who use these tools are in good faith, they must pay attention to potential problems deriving from these aspects. The research community is spending a lot of efforts in order to mitigate the mentioned risks and to provide guidelines for the adoption of such systems [25, 123].

While I cannot fully prevent such uses once our models are made public, I do hope that writing about risks explicitly and also raising awareness of this possibility in the general public are ways to contain the effects of potential harmful uses.

The contributors of human judgements elicited for the works reported in this thesis have been fairly compensated.

Chapter 2

Natural Language Processing: a brief overview

In this chapter, we will introduce Natural Language Processing (NLP). A particular focus is dedicated to deep learning techniques for NLP since recently they became very popular. Part of this chapter is inspired by [115], who provides an extensive overview of deep learning for NLP, however section 2.8 reports more recent advancements of deep learning for NLP.

2.1 What is NLP about

"NLP is the field of designing methods and algorithms that take as input or produce as output unstructured, natural language data" [115]. What makes NLP tough is that human languages are highly ambiguous, variable and they evolve through time.

Humans are great at producing and understanding language although they are very poor at formally understanding and describing the rules that govern language. Indeed understanding and producing language using computers is thus highly challenging. Besides the challenges of dealing with ambiguous and variable inputs in a system with an ill-defined and unspecified set of rules, natural language exhibits an additional set of properties that make it even more challenging for computational approaches: it is (i) discrete, (ii) compositional, and (iii) sparse.

Languages are symbolic and discrete. The basic elements of written language are characters. Characters form words that in turn denote objects, concepts, events, actions, and ideas. Both characters and words are discrete symbols: words evoke in us a certain mental representation, but they are also distinct symbols, whose meaning is external to them. There is no inherent information about the words that can be inferred from the symbols themselves, or from the individual letters they are made of.

Languages are compositional. Letters form words, and words form phrases and sentences. The meaning of a phrase can be larger than the meaning of the individual words that comprise it and follows a set of intricate rules. In order to interpret a text, we thus need to work beyond the level of letters and words and look at long sequences of words such as sentences, or even complete documents.

Languages are sparse. There is an infinite way to combine characters and words to form meanings.

The researchers started working on NLP in early 1950 focusing on machine translation (MT) [146]. Different approaches have been applied to solve several NLP tasks but the best results have been obtained recently by adopting machine learning techniques [115]. Even more recently deep learning techniques provided encouraging results. In the scope of this thesis, we will consider machine learning and deep learning techniques only. Different approaches are not described in this chapter. Before diving into NLP, a brief introduction to machine learning and neural networks is provided in the next section.

2.2 Machine Learning, Neural Network and Deep Learning

Using Mitchell's words [215] *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."*. Then Machine Learning is the study of computer algorithms that improve automatically through experience. Machine learning approaches are traditionally divided into two broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

- Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.
- Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

In NLP most frequently supervised learning techniques are used. Supervised machine learning theory is a very large topic, and it is not discussed here. Complete treatments of this topic, such as [276], are available.

Neural networks were initially inspired by the brain's computation mechanism, which consists of computation units called neurons. While the connections between artificial neural networks and the brain are in fact rather slim, we repeat the metaphor here for completeness. In the metaphor, a neuron is a computational unit that has scalar inputs and outputs. Each input has an associated weight. The neuron multiplies each input by its weight, and then sums them, applies a nonlinear function to the result, and passes it to its output. Figure 2.1 represents such a metaphor.

Neural networks can be defined by a directed graph, where the vertices represent the neurons and the edges represent the connections between them. Each neuron is associated with an activation function and each connection has a weight value.

One of the most common types of neural networks are the Feed-forward Neural Networks (FFNNs): a class of NNs in which the information is propagated from the input layer towards the output layer without feedback connections. They are able to process flat data such as fixed-sized vectors of variables. Figure 2.2 shows the architecture of an FFNN with a single hidden layer. FFNN are typically trained to exploit gradient-based techniques such as Back Propagation [263].

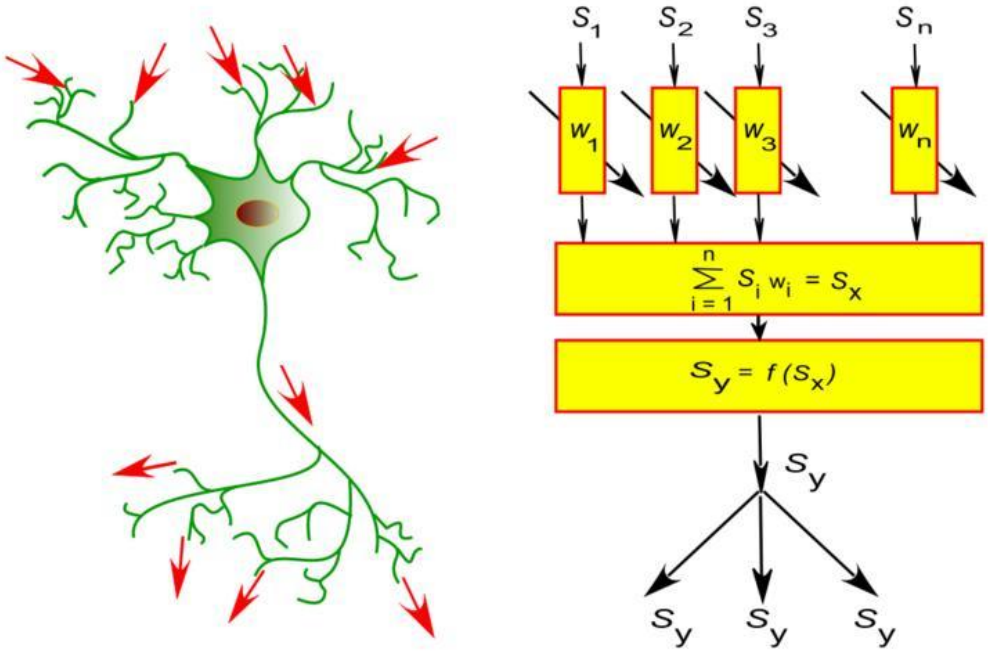


FIGURE 2.1: Natural and Artificial Neuron

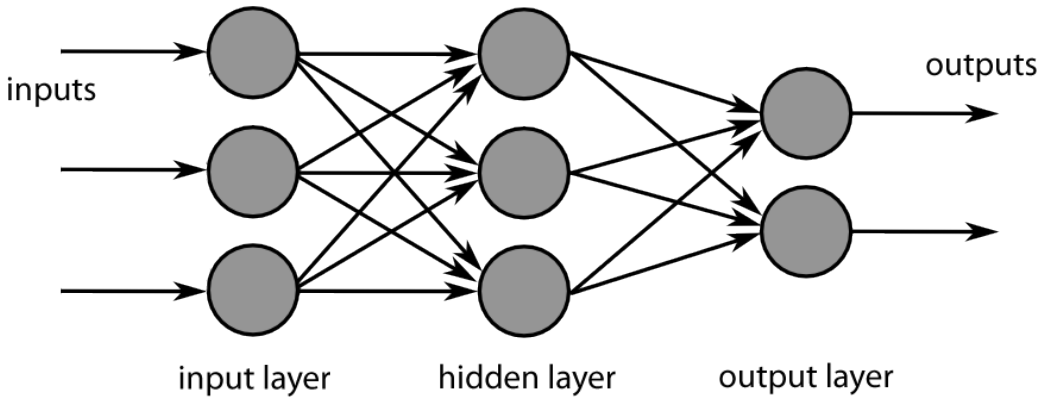


FIGURE 2.2: An example of Feed-forward Neural Network

Deep learning is part of the broader family of neural networks. The name deep-learning stems from the fact that many layers are chained together. While all of the machine learning can be characterized as learning to make predictions based on past observations, deep learning approaches work by learning to not only predict but also to correctly represent the data, such that it is suitable for prediction. Given a large set of desired input-output mapping, deep learning approaches work by feeding the data into a network that produces successive transformations of the input data until a final transformation predicts the output. The transformations produced by the network are learned from the given input-output mappings, such that each transformation makes it easier to relate the data to the desired label.

2.3 ML for NLP: the representation problem

Machine learning research provides us with a large set of models and algorithms for training them. Most of these models take as input vectors **x** and produce predictions. In NLP, the vectors **x** are derived from textual data, in order to reflect various linguistic properties of the text. The mapping from textual data to real-valued vectors is called *feature extraction* or *feature representation* and is done by a feature function. Deciding on the right features is an integral part of a successful machine learning project and this is especially true for language data, which comes in the form of a sequence of discrete symbols. This sequence needs to be converted somehow to a numerical vector, in a non-obvious way.

In the following subsections, we will review the most common approach to perform feature extraction on textual data. As words and letters are discrete items, features often take the form of indicators or counts. An indicator feature takes a value of 0 or 1, depending on the existence of a condition. A count takes a value depending on the number of times some event occurred.

2.3.1 Lexical features

When our focus entity is a word outside of a context, our main source of information is the letters comprising the word and their order, as well as properties derived from these, such as the length of the word, the orthographic shape of the word (Is the first letter capitalized? Are all letters capitalized? Does the word include a hyphen? Does it include a digit? And so on), and prefixes and suffixes of the word. We may also look at the word with relation to external sources of information: How many times does the word appear in a large collection of text? Does the word appear in a list of common person names? And so on.

We often look at the lemma (the dictionary entry) of the word, mapping forms such as *"booking"*, *"booked"*, *"books"* to their common lemma *"book"*. This mapping is usually performed using lemma lexicons or morphological analyzers, which are available for many languages.

An additional source of information about word forms are lexical resources. These are essentially dictionaries that are meant to be accessed programmatically by machines rather than read by humans. A lexical resource will typically contain information about words, linking them to other words and/or providing additional information. For example, for many

languages, there are lexicons that map inflected word forms to their possible morphological analyses (i.e., telling you that a certain word may be either a plural feminine noun or a past-perfect verb). Such lexicons will typically also include lemma information. A very well-known lexical resource is WordNet [93]. WordNet is a large manually curated dataset attempting to capture conceptual semantic knowledge about words. Another famous lexical resource is FrameNet [96]: a manually curated lexical resource that focus around verbs, listing for many verbs the kinds of argument they may take.

When we consider a sentence, a paragraph, or a document, the observable features are the counts and the order of the letters and the words within the text. A very common feature extraction procedure for sentences and documents is the **bag-of-words** approach (BOW). In this approach, we look at the histogram of the words within the text, i.e., considering each word count as a feature. When using the bag-of-words approach, it is common to use TF-IDF weighting [192, 260]. Besides words, also consecutive pairs or triplets of words could be considered. These are called ngrams and they are largely adopted for feature extraction.

A standard feature to measure the lexical variety of a text is constituted by the Type/Token Ratio (TTR), which refers to the ratio between the number of lexical types and the number of tokens within a text.

2.3.2 Raw text features

Some other non-lexical features can be extracted from raw texts:

- document length: length of the document calculated both in terms of the total number of tokens and of the total number of sentences it is constituted of;
- sentence length: average length of sentences in a text or collection of texts, calculated as the average number of tokens per sentence;
- word length: calculated as the average number of characters per word.

Sentence length and word length are typically seen as proxies of syntactic complexity and lexical complexity respectively [38].

2.3.3 Inferred linguistic features

Sentences in natural language have structures beyond the linear order of their words. While the exact structure of language is still a mystery, and rules governing many of the more intricate patterns are either unexplored or still open for debate among linguists, a subset of phenomena governing language are well documented and well understood.

These include concepts such as word classes (part-of-speech tags), morphology, syntax, and even parts of semantics. While the linguistic properties of a text are not observable directly from the surface forms of words in sentences and their order, they can be inferred from the sentence string with varying degrees of accuracy. Specialized systems exist for the prediction of parts of speech, syntactic trees, semantic roles, discourse relations, and other linguistic properties with various degrees of accuracy, and these predictions often serve as good features for further classification problems.

Linguistic profiling

By relying on different levels of linguistic annotation, it is possible to extract a large number of features modelling lexical, grammatical and semantic phenomena that, all together, contribute to characterize language variation within and across texts. These are the prerequisites of linguistic profiling. Following this approach, the linguistic structure of a text is analyzed to extract relevant features, and a representation of the text is constructed out of occurrence statistics of these features, be they absolute or relative frequencies or more complex statistics. In linguistic profiling, each text or collection of texts is thus assigned a feature-based representation covering different levels of linguistic description [38]. Those approaches let us to extract meta-knowledge from texts [70], namely, what are the features and how they combine together within a specific language variety as opposed to another one of the same nature, be it determined on the basis of the communicative purposes in a given situational context, or of the speaker socio-demographic traits, or of the author, or of the addressee. Meta-knowledge extraction thus consists in associating the feature-based representation of a (collection of) text(s) with a functional context, or with a class of speakers and/or addressees, or with individual authors. [240] shows how a style analysis can distinguish hyperpartisan news from the mainstream, and satire from both.

2.3.4 Distributional Features

Until now we treated words as discrete and unrelated symbols. We did achieve some form of generalization across word types by mapping them to coarser-grained categories such as parts-of-speech; generalizing from inflected words forms to their lemmas; looking at membership in lists or dictionaries; or looking at their relation to other words using lexical resources such as WordNet. However, these solutions are quite limited: they either provide very coarse grained distinctions, or otherwise rely on specific, manually compiled dictionaries.

The distributional hypothesis of language [97, 129] states that the meaning of a word can be inferred from the contexts in which it is used. Many algorithms were derived over the years to make use of this property, and learn generalizations of words based on the contexts in which they occur. These can be broadly categorized into clustering-based methods, which assign similar words to the same cluster and represent each word by its cluster membership [34, 213], and to embedding-based methods which represent each word as a vector such that similar words have similar vectors [66, 211]. We will dive more into word embeddings methods and the use of word vectors in section 2.6.

2.4 Linguistic Annotation Pipelines

Almost any of the mentioned feature extraction techniques mentioned in the previous section requires at least a basic pre-processing of textual data. In this section, we will describe the linguistic annotation pipelines which are tools designed to pre-process textual data. A pipeline is typically made by a subset of the following modules:

- Sentence Splitter: a module that splits the documents in a sequence of sentences;

- Tokenizer: a module that splits sentences in a sequence of words (or more precisely tokens);
- Part of Speech Tagger: a module that classifies each token into one morphosyntactic category;
- Lemmatizer or Stemmer: a module that derives for each token its base form;
- Morphological Tagger: a module that derives for each token its morphological features;
- Syntactic Parser: a module that build the syntactic tree of each sentence.

Several open source lingustic pipeline such as SpaCy[1] and UDPipe[2] are available.

2.4.1 Sentence Splitting

Segmenting a text into sentences is generally based on punctuation. This is because certain kinds of punctuation (periods, question marks, exclamation points) tend to mark sentence boundaries. Question marks and exclamation points are relatively unambiguous markers of sentence boundaries. Periods, on the other hand, are more ambiguous. The period character '.' is ambiguous between a sentence boundary marker and a marker of abbreviations like *"Mr."* or *"Inc."* The previous sentence that you just read showed an even more complex case of this ambiguity, in which the final period of Inc. marked both an abbreviation and the sentence boundary marker. In general, sentence splitting methods work by building a binary classifier (based on a sequence of rules or on machine learning) which decides if a period is part of the word or is a sentence boundary marker. In making this decision, it helps to know if the period is attached to a commonly used abbreviation; thus an abbreviation dictionary is useful.

2.4.2 Tokenization

Word tokenization may seem very simple in languages like English and Italian that separate words via a special 'space' character. However not every language does this (Chinese, Japanese, and Thai, for example, do not). Moreover the white-space is not sufficient by itself, indeed punctuation and other phenomena such as hyphens and apostrophe need to be handled. For languages such as English and Italian complex rule-based systems are typically used for tokenization, while for other languages statistical approaches are adopted.

Tokenizing on white-space and punctuation has been widely adopted, however, in this case, the definition of a word is quite technical: it is derived from the way things are written. Another common definition from linguistic theory takes a word to be "the smallest unit of meaning." By following this logic, the whitespace-based definition is problematic because a single token might be a compound word (eg ice cream) or more frequently can contain morphological information in the form of suffixes or affixes (eg the final 's' in the third person singular). This is especially frequent in morphologically rich languages such as Italian. Moreover in some languages such as German compound words are extremely frequent.

[1] https://spacy.io/

[2] http://ufal.mff.cuni.cz/udpipe

Considering sub-tokens splitting might help in reduce data sparseness and in processing more efficiently rare words.

In general, we distinguish between words and tokens. We refer to the output of a tokenizer as a token, and to the meaning-bearing units as words. A token may be composed of multiple words, multiple tokens can be a single word, and sometimes different tokens denote the same underlying word.

In the last few years with the advent of Transformers architecture (which will be described in section 2.8) different tokenization approaches that consider sub-token splitting has been widely adopted. Those approaches (such as [274]) relies on variants of an old compression algorithm called Byte Pair Encoding (BPE) [103]. Those algorithms are trained on huge text corpora with no annotation required. Thanks to its mathematical properties, BPE brings a good balance between character- and word-level hybrid representations which reduce data sparseness issues. This behaviour also enables the encoding of any rare words in the vocabulary with appropriate sub-word tokens without introducing any "unknown" tokens. This especially applies to languages like German and Italian.

2.4.3 Part of Speech Tagging

Part of Speech (POS) Tagging is the process of marking up each token in a text as corresponding to a particular part of speech, based on both its definition and its context. In linguistics, a POS is a category of words that have similar grammatical properties. Words that are assigned to the same part of speech generally display similar syntactic behaviour and sometimes similar morphology in that they undergo inflexion for similar properties. Commonly listed English parts of speech are noun, verb, adjective, adverb, pronoun, preposition, conjunction, interjection, numeral, article, or determiner. POS tagging is typically done exploiting machine learning techniques.

2.4.4 Lemmatisation or stemming

Lemmatisation in linguistics is the process of grouping together the inflected forms of a word so that they can be analysed as a single item, identified by the word's lemma, or dictionary form. An alternative to lemmatisation is stemming: the process of reducing inflected or derived words to their word stem, base or root form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Unlike stemming, lemmatisation depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighboring sentences or even an entire document. Lemmatisation is typically done by exploiting curated lexicons. In general stemming is an easier tasks than lemmatisation, but it is less reliable, especially for morphologically rich languages such as Italian.

2.4.5 Syntactic Parsing

Syntactic parsing is the process of determining the syntactic structure of a text by analyzing its constituent words based on an underlying grammar. There are mainly two ways to represent the syntax of a text: constituency and dependency parsing.

The constituency parse tree is based on the formalism of context-free grammars. In this type of tree, the sentence is divided into constituents, that is, sub-phrases that belong to a specific category in the grammar. As opposed to constituency parsing, dependency parsing does not make use of phrasal constituents or sub-phrases. Instead, the syntax of the sentence is expressed in terms of dependencies between words that is, directed, typed edges between words in a graph. More formally, a dependency parse tree is a graph $G = (V, E)$ where the set of vertices V contains the words in the sentence, and each edge in E connects two words. The graph must satisfy three conditions:

- There has to be a single root node with no incoming edges.
- For each node v in V, there must be a path from the root R to v.
- Each node except the root must have exactly one incoming edge.
- Additionally, each edge in E has a type, which defines the grammatical relation that occurs between the two words.

Figure 2.3 reports examples of constituency and dependency parse trees. During the last years, dependency parsing is the most adopted approach. Typically the parsing problem is reduced to a classification problem exploiting transitions algorithms [308, 224] and/or graph-based algorithms [64, 47, 206]. The classification is then typically done exploiting machine learning techniques.

The Universal Dependencies (UD) project[3] provides consistent annotation of grammar (parts of speech, morphological features, and syntactic dependencies) across different human languages. UD is an open community effort with over 300 contributors producing more than 150 treebanks in 90 languages.

2.5 Language Modelling

Language modelling is the task of assigning a probability to sentences in a language. From a mathematical point of view, this problem can be equivalently reformulated as assigning a probability for the likelihood of a given word (or a sequence of words) to follow a sequence of words.

Perfect performance at the language modelling task, namely predicting the next word in a sequence with a number of guesses that is the same as or lower than the number of guesses required by a human participant, is an indication of human-level intelligence. Even without achieving human-level performance, language modelling is a crucial component in real-world

[3] https://universaldependencies.org/

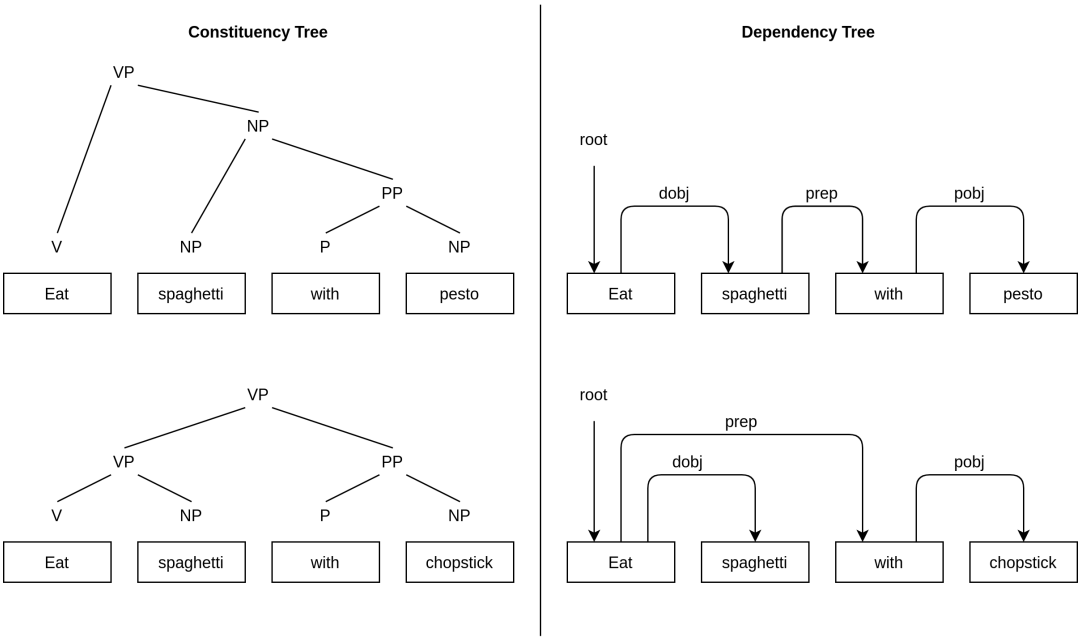


FIGURE 2.3: Examples of Constituency and Dependency parse trees

applications such as machine-translation and automatic speech recognition, where the system produces several translations or transcription hypotheses, which are then scored by a language model. For these reasons, language modeling plays a central role in NLP research. Language models can also be used for generating sentences (this topic will be widely discussed in chapter 3).

Formally, the task of language modeling is to assign a probability to any sequence of words $w_{1:n}$, i.e., to estimate $P(w_{1:n})$. Using the chain rule of probability, this can be rewritten as:

$$P(w_1)P(w_2|w_1)P(w_3|w_{1:2})...P(w_n|w_{1:n-1}) \tag{2.1}$$

While the task of modelling a single word based on its left context seems more manageable than assigning a probability score to an entire sentence, the last term in the equation still requires conditioning on $n-1$ words, which is as hard as modelling an entire sentence. For this reason, language models make use of the Markov-assumption, stating that the future is independent of the past given the present. More formally, a kth order Markov-assumption assumes that the next word in a sequence depends only on the last k words:

$$P(w_{i+1}|w_{1:i}) \approx P(w_{i+1}|w_{i-k:i}) \tag{2.2}$$

Then estimating the probability of sentences becomes:

$$P(w_{1:n}) \approx \prod_{i=1}^{n} P(w_i|w_{i-k:i-1}) \tag{2.3}$$

While the kth order Markov assumption is clearly wrong from a linguistic point of view, it still produces strong language modelling results for relatively small values of k , and it has

been the dominant approach for language modelling for decades.

Those traditional model are easy to train, scale to large corpora and work well in practice. However, they have several important shortcomings. Firstly the Markov assumption is wrong, several smoothing techniques has been adopted to mitigate this issue, but still the theoretical issue is there. Moreover, those language models suffer from lack of generalization across contexts: having observed *black car* and *blue car* does not influence our estimates of the event *red car* if we haven't seen them before.

Neural network language models such as [220, 26] solve some of the shortcomings of traditional language models: they allow conditioning on increasingly large context sizes with only a linear increase in the number of parameters, alleviate the need for manually designing backoff orders, and support generalization across different contexts.

There are several metrics for evaluating language modelling. The application-centric ones evaluate them in the context of performing a higher-level task, for example by measuring the improvement in translation quality when switching the language-modelling component in a translation system from model A to model B. A more intrinsic evaluation of language models is using perplexity over unseen sentences. Perplexity is an information theoretic measure of how well a probability model predicts a sample. Low perplexity values indicate a better fit. Given a text corpus of n words $w_{1:n}$ (n can be in the millions) and a language model function LM assigning a probability to a word based on its history, the perplexity of LM with respect to the corpus:

$$2^{-\frac{1}{n}\sum_{n}^{i=1} log_2(w_i|w_{1:i-1})} \tag{2.4}$$

Good language models will assign high probabilities to the events in the corpus, resulting in lower perplexity values.

2.6 Pre-trained Word Embeddings

An embedding is a mapping of a discrete categorical variable to a vector of continuous numbers. In the context of neural networks, embeddings are low-dimensional, learned continuous vector representations of discrete variables. Embeddings are useful because they can reduce the dimensionality of categorical variables and meaningfully represent categories in the transformed space.

Neural network embeddings overcome the two limitations of the traditional indicators and counts representation method because:

- embedding vector are dense and their dimensionality does not increase with the cardinality of the categories;
- similar categories can be represented with similar vectors.

In NLP embeddings can be used to represent the meaning of words, but how do we train them? When enough supervised training data is available, one can just treat the feature embeddings the same as the other model parameters: initialize the embedding vectors to

random values, and let the network-training procedure tune them into *good* vectors. However the common case is that the annotated data available are not enough to learn *good* vectors. In such cases the embeddings are learned on auxiliary tasks, that can be trained on huge amounts of unannotated text. Then the learned embeddings can be used for the main task. That is why we refer to them as pre-trained word embeddings.

The key idea behind the word embeddings training methodologies is that one would like the embedding vectors of "similar" words to have similar vectors. While word similarity is hard to define and is usually very task-dependent, the current approaches derive from the distributional hypothesis [97, 129] already mentioned in section 2.3.4, stating that words are similar if they appear in similar contexts. The different training methods such as [66, 67, 211, 210] all create supervised training instances in which the goal is to either predict the word from its context or predict the context from the word, which are indeed the goal language modelling. Word2Vec [211, 210] is one of the most widely used family of algorithms and it were inspired by this property of language modelling. The Word2Vec algorithms are designed to perform the same side effects as language modelling, using a very efficient and flexible framework.

An important benefit of training word embeddings on large amounts of unannotated data is that it provides vector representations for words that do not appear in the supervised training set. For the Italian language two huge general-purpose corpora are available [15, 230], but for task on specific language domain the availability of in-domain data might be useful, for example [19] provides a huge corpus of Twitter posts.

2.6.1 Limitations

The distributional hypothesis offers an appealing platform for deriving word similarities by representing words according to the contexts in which they occur. It does, however, have some inherent limitations that should be considered when using the derived representations.

Black Sheep

When using texts as the conditioning contexts, many of the more "trivial" properties of the word may not be reflected in the text, and thus not captured in the representation. this happens because of a well-documented bias in people's use of language, stemming from efficiency constraints on communication: people are less likely to mention known information than they are to mention novel one. Thus, when people talk of white sheep, they will likely refer to them as sheep, while for black sheep they are much more likely to retain the colour information and say black sheep. A model trained on text data only can be greatly misled by this.

Antonyms Words

Antonyms words that are the opposite of each other (good vs. bad) tend to appear in similar contexts. As a consequence, models based on the distributional hypothesis tend to judge antonyms as very similar to each other.

Corpus Biases

The distributional methods reflect the usage patterns in the corpora on which they are based, and the corpora in turn reflect human biases in the real world such as stereotypes, racism, gender biases and so on [144, 30].

Lack of Context

The word embedding represent words independently by their context. From a linguistic point of view does not exist a context-independent meaning for a word [98]. An obvious manifestation of this is the case of polysemy: some words have obvious multiple senses, for example a *"bank"* may refer to a financial institution or to the side of a river. A single vector for all the semantic forms of a word might be not representative.

2.7 Neural Networks Architectures for NLP

Classic feed-forward neural networks are general-purpose architectures: nothing in them is tailored specifically for language data. In literature are available some neural architectures that are more specialized for dealing with language data. In the following subsection, we will describe two commonly used architecture for NLP: (i) Convolutional Neural Network (CNN) [173] and (ii) Recurrent Neural Network (RNN) [91]. Indeed textual data are naturally represented as sequences of words: this kind of neural networks take as input ordered sequences of features vectors. Typically those architectures take as input sequences of word emebeddings.

2.7.1 Convolutional Neural Networks

For certain NLP tasks, some of the words in sentences are more informative than others. Consider for example sentiment analysis some words like *good*, *bad*, *awesome*, *awful* might be very informative and to a good approximation, an informative clue is informative regardless of its position in the sentence. A possible approach could be to feed all of the sentence words into a learner, and let the training process Figure out the important clues. One possible solution is feeding a BOW representation into an FFN. However, a downside of the BOW approach is that it ignores the ordering information completely. The following two sentences would be represented equally, while they have an opposite sentiment:

- "it was not good, it was actually quite bad"
- "it was not bad, it was actually quite good"

While the global positions of the indicators "not good" and "not bad" do not matter for the classification task, the local ordering of the words is very important. Looking at n-grams is much more informative than looking at a bag-of-words: a naive approach would suggest embedding word-pairs (bi-grams) or word-triplets (tri-grams) rather than words and building a BOW over the embedded n-grams. While such an architecture is indeed quite effective, it

will result huge embedding matrices, will not scale for longer n-grams, and will suffer from data sparsity problems as it does not share statistical strength between different n-grams.

A common approach adopted to tackle those issues is the usage of CNNs: neural networks designed to identify indicative local predictors in a large structure, and to combine them to produce a fixed-size vector representation of the structure, capturing the local aspects that are most informative for the prediction task at hand. The convolutional architecture also allows to share predictive behavior between n-grams that share similar components, even if the exact n-gram was never seen at test time. CNN evolved in the computer vision community, where they showed great success as object detectors. When applied to images, the architecture is using 2D convolutions. When applied to text, we are mainly concerned with 1D convolutions. [67] introduced CNNs to the NLP community.

The main idea behind the CNNs architecture for language tasks is to apply a non-linear function over each instantiation of a k-word sliding window over the sentence. This function (also called "filter") transforms a window of k words into a scalar value. Several filters can be applied, resulting in l dimensional vector (each dimension corresponding to a filter) that captures important properties of the words in the window. Then, a *pooling* operation is used to combine the vectors resulting from the different windows into a single l-dimensional vector, by taking the max or the average value observed in each of the l dimensions over the different windows. The intention is to focus on the most important "features" in the sentence, regardless of their location: each filter extracts a different indicator from the window, then the pooling operation zooms in on the important indicators. The resulting l-dimensional vector is then fed further into a network that is used for prediction. The gradients that are propagated back from the network's loss during the training process are used to tune the parameters of the filter function to highlight the aspects of the data that are important for the task the network is trained for. Intuitively, when the sliding window of size k is run over a sequence, the filter function learns to identify informative k-grams. Figure 2.4 reports the architecture of a CNN for an NLP task.

The 1D convolution approach described can be thought of as an n-gram detector. A convolution layer with a window of size k is learning to identify indicative k-grams in the input. The approach can be extended into a hierarchy of convolutional layers, in which a sequence of convolution layers are applied one after the other. For r layers with a window of size k, the last layer output vector will be sensitive to a window of $r(k-1)$ words. Moreover, the vector output vector can be sensitive to gappy-ngrams such as "not ... good" where "..." stands for a short sequence of words.

2.7.2 Recurrent Neural Networks

RNNs allow representing arbitrarily sized sequential inputs in fixed-size vectors while paying attention to the structured properties of the inputs. For this reason, they are arguably the strongest contribution of deep-learning to the statistical natural-language processing tool-set.

We use $\mathbf{x}_{i:j}$ to denote a sequence of vectors $\mathbf{x}_i, ..., \mathbf{x}_j$. On a high-level, the RNN is a function that takes as input an arbitrary length ordered sequence of n d_{in}-dimensional vectors $\mathbf{x}_{1:n}(\mathbf{x}_i \in \mathbb{R}^{d_{in}})$ and returns as output a single d_{out} dimensional vector $\mathbf{y_n} \in \mathbb{R}^{d_{out}}$:

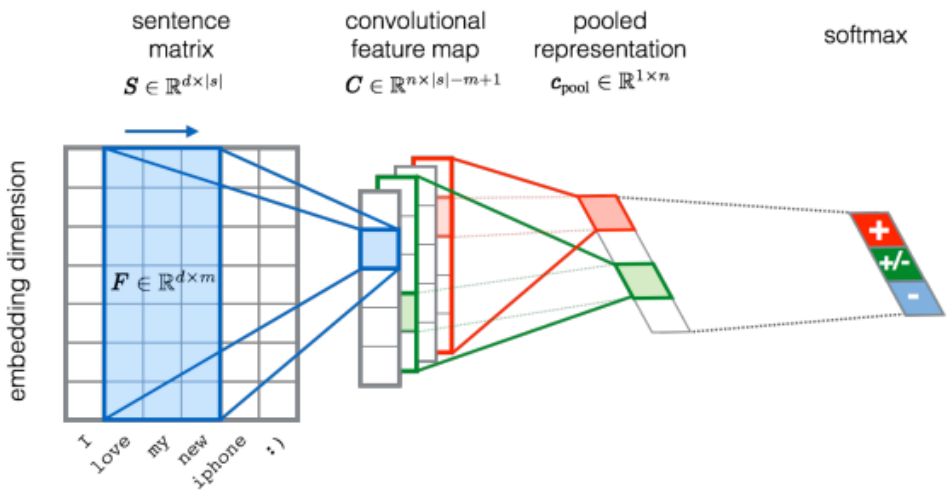


FIGURE 2.4: The architecture of a CNN for sentiment classification. Image source: [275]

$$\mathbf{y_n} = RNN(x_{1:n}) \tag{2.5}$$

This implicitly defines an output vector $\mathbf{y}_i$ for each prefix $\mathbf{x}_{1:i}$ of the sequence $\mathbf{x}_{1:n}$. We denote by RNN^* the function returning this sequence:

$$\begin{aligned} \mathbf{y_{1:n}} &= RNN^*(x_{1:n}) \\ \mathbf{y_i} &= RNN(x_{1:i}) \end{aligned} \tag{2.6}$$

The output vector $\mathbf{y}_n$ is then used for further prediction. Looking in a bit more detail, the RNN is defined recursively, by means of a function R taking as input a state vector $\mathbf{s}_{i-1}$ and an input vector $\mathbf{x}_i$ and returning a new state vector $\mathbf{s}_i$. The state vector $\mathbf{s}_i$ is then mapped to an output vector $\mathbf{y}_i$ using a simple deterministic function $O(\cdot)$. The base of the recursion is an initial state vector, $\mathbf{s}_0$, which is also an input to the RNN. For brevity, we often omit the initial vector $\mathbf{s}_0$, or assume it is the zero vector. When constructing an RNN, much like when constructing a feed-forward network, one has to specify the dimension of the inputs $\mathbf{x}_i$ as well as the dimensions of the outputs $\mathbf{y}_i$. The dimensions of the states $\mathbf{s}_i$ are a function of the output dimension.

$$\begin{aligned} RNN^*(\mathbf{x}_{1:n}, \mathbf{s}_0) &= \mathbf{y}_{1:n} \\ \mathbf{y}_i &= O(\mathbf{s}_i) \\ \mathbf{s}_i &= R(\mathbf{s}_{i-1}, \mathbf{x}_i) \end{aligned} \tag{2.7}$$

RNN network are trained with a procedure called backpropagation through time (BPTT) [305]. Note that the RNN does not do much on its own, but serves as a trainable component in a larger network. The final prediction and loss computation are performed by that larger

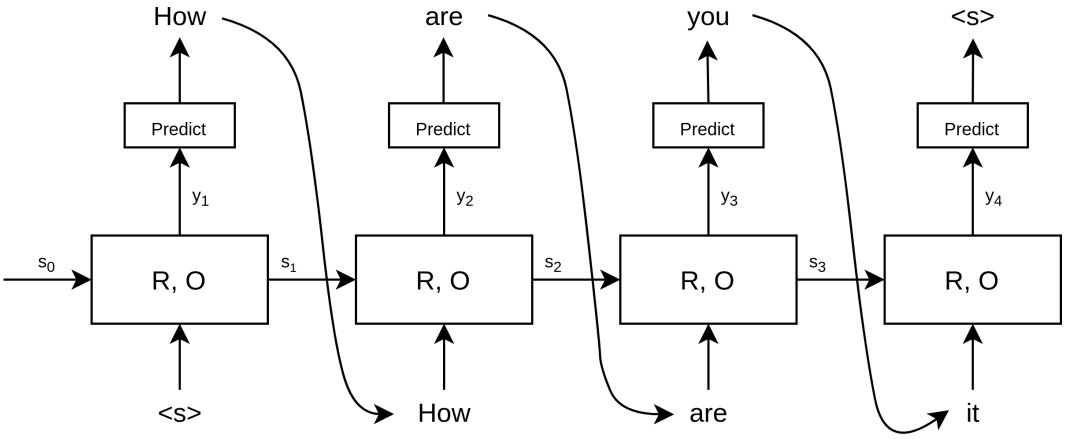


FIGURE 2.5: RNN used as generator

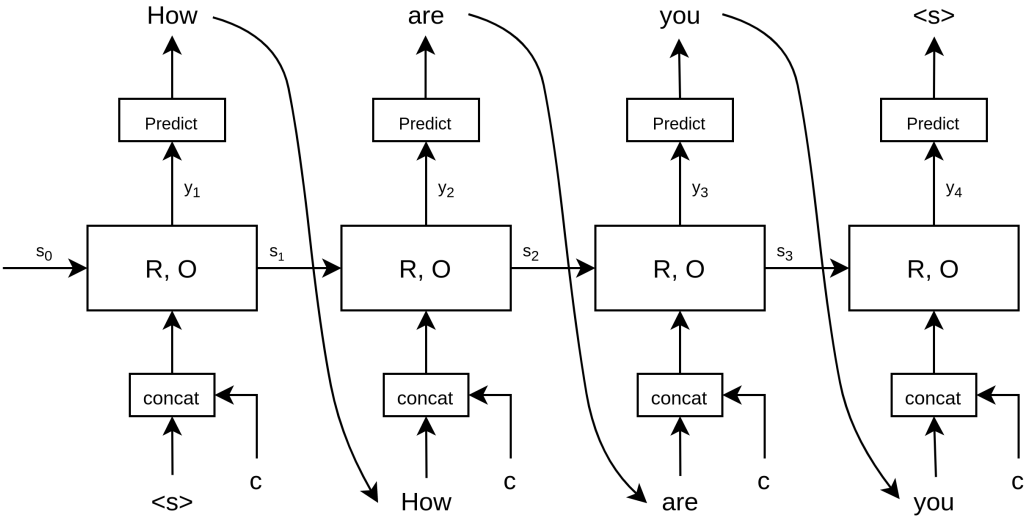


FIGURE 2.6: RNN used as generator with conditioning vector

network, and the error is back-propagated through the RNN. This way, the RNN learns to encode properties of the input sequences that are useful for the further prediction task. The supervision signal is not applied to the RNN directly, but through the larger network.

The RNN function provides a framework for conditioning on the entire history $\mathbf{x}_{1:i}$ without resorting to the Markov assumption which is traditionally used for modelling sequences. Indeed, RNN based language models result in very good perplexity scores when compared to n-gram based models. A special case of using the RNN architecture for language modeling is sequence generation. Sequence generation works by tying the output of the transducer at time i with its input at time $i+1$: after predicting a distribution over the next output symbols $p(t_i = k|t_{1:i1})$, a token t_i is chosen and its corresponding embedding vector is fed as the input to the next step. The process stops when generating a special end-of-sequence symbol. See Figure 2.5 for a graphical representation.

The sequence generation can be conditioned by concatenating to each word vector in input a context vector, see Figure 2.6. The context vector can have many forms. For examples it can be the output vector encoded by another RNN cell (see Figure 2.7), it is the case of

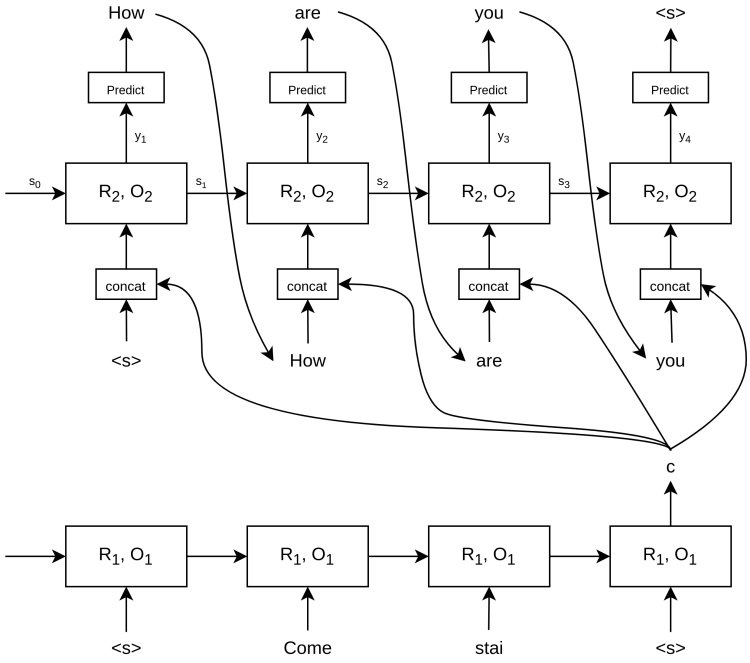


FIGURE 2.7: Sequence to sequence model

sequence to sequence models [51]. This kind of models are very powerful as it can be used to generate a sequence of words starting from a sequence of words and this applies to tasks such as machine translation or conversational systems.

Bidirectional Recurrent Neural Networks

A useful elaboration of an RNN is a bidirectional-RNN (biRNN) [270]. Consider the task of sequence tagging over a sentence $\mathbf{x}_{1:n}$. An RNN allows us to compute a function of the ith word x_i based on the past—the words $\mathbf{x}_{1:i}$ up to and including it. However, the following words $\mathbf{x}_{i+1:n}$ may also be useful for prediction. Much like the RNN relaxes the Markov assumption and allows looking arbitrarily back into the past, the biRNN relaxes the fixed window size assumption, allowing to look arbitrarily far at both the past and the future within the sequence.

Gated Recurrent Neural Networks

Despite the nice properties described above is important to underline that RNNs are very hard to train effectively because of the vanishing gradients problem [232]. Error signals in later steps in the sequence diminish quickly in the backpropagation process, and do not reach earlier input signals, making it hard for the RNN to capture long-range dependencies. Gating-based architectures, such as the LSTM [135] and the GRU [50] are designed to solve this deficiency. In NLP practice those architectures are used.

Attention mechanism

Sequence to sequence models encodes the input sentence is encoded into a single vector, which is then used as a conditioning context for a generator. These architectures force the encoded vector $c = RNN^{enc}(\mathbf{x}_1 : n)$ to contain all the information required for generation and require the generator to be able to extract this information from the fixed-length vector. Given these rather strong requirements, the architecture works surprisingly well. However, in many cases it can be substantially improved by the addition of an attention mechanism. The conditioned generation with attention architecture [8] relaxes the condition that the entire source sentence is encoded as a single vector. Instead, the input sentence is encoded as a sequence of vectors, and the decoder uses a soft attention mechanism in order to decide on which parts of the encoding input it should focus. The encoder, decoder, and attention mechanism are all trained jointly in order to play well with each other.

The encoder-decoder with attention architecture encodes a length n input sequence $\mathbf{x}_{1:n}$ using a biRNN, producing n vectors $\mathbf{c}_{1:n}$:

$$\mathbf{c}_{1:n} = ENC(\mathbf{c}_{1:n}) = biRNN^*(\mathbf{x}_{1:n}) \tag{2.8}$$

The decoder can then use these vectors as a read-only memory representing the conditioning sentence: at any stage j of the generation process, it chooses which of the vectors $\mathbf{c}_{1:n}$ it should attend to, resulting in a focused context vector $\mathbf{c}^j = attend(\mathbf{c}_{1:n}, \hat{t}_{1:j})$. The focused context vector $\mathbf{c}^j$ is then used for conditioning the generation at step j.

The $attend(\cdot, \cdot)$ is a parametrized function, its parameteres are learned during training time. The attention mechanism is soft, meaning that at each stage the decoder sees a weighted average of the vectors $\mathbf{c}_{1:n}$, where the weights are chosen by the attention mechanism. More formally, at stage j the soft attention produces a mixture vector $\mathbf{c}^j$:

$$\mathbf{c}^j = \sum_{i=1}^{n} \alpha^j_{[i]} \cdot \mathbf{c}_i \tag{2.9}$$

where $\alpha^j \in \mathbb{R}^n_+$ is the attention vector for stage j, whose elements $\alpha^j_{[i]}$ are all positive and sum to one. The values $\alpha^j_{[i]}$ are produced in a two stage process: first, unnormalized attention weights $\alpha^{-j}_{[i]}$ are produced using a FFN FFN^{att} taking into account the decoder state at time j and each of the vectors $\mathbf{c}_i$:

$$\alpha^{-j} = FNN^{att}([\mathbf{s}_j], [\mathbf{c}_1]), ..., FNN^{att}([\mathbf{s}_j], [\mathbf{c}_n]) \tag{2.10}$$

The unnormalized weights α^{-j} are then normalized into a probability distribution using the softmax function:

$$\alpha^j = softmax(\alpha^{-j}) \tag{2.11}$$

Figure 2.8 reports a diagram of the sequence to sequence with the attention model.

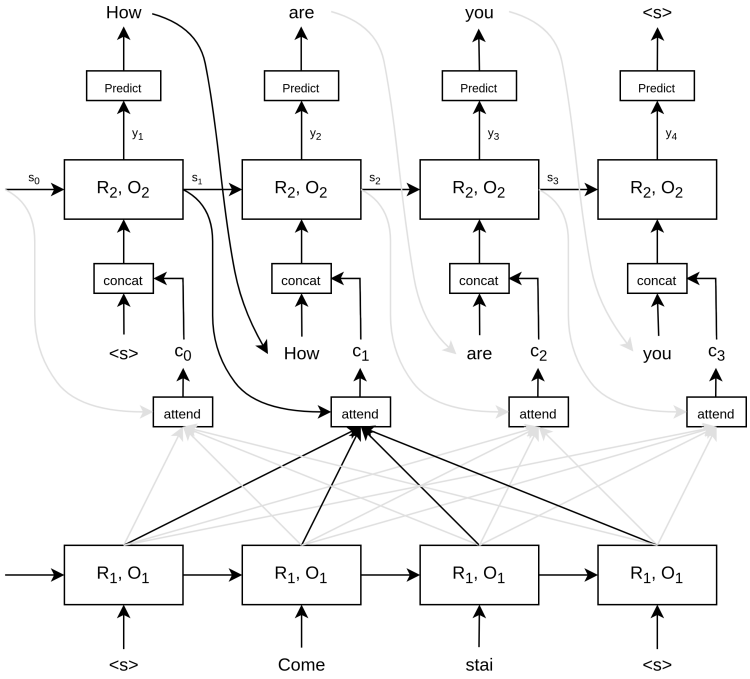


FIGURE 2.8: Sequence to sequence with attention model

2.8 Transformers

[297] introduces a novel architecture called Transformer which exploits the attention mechanism we saw earlier. In particular, the Transformer architecture eschews recurrence and relays entirely on an attention mechanism to draw global dependencies between input and output. This property let Transformers allow for significantly more parallelization at the cost of quadratic complexity in the input sequence length. Their performances represent the state of the art for a lot of tasks.

Sequence to sequence models encode an input sequence of symbol representations $\mathbf{x}_{1:n}$ to a sequence of continuous representations $\mathbf{z}_{1:n}$. Given $\mathbf{z}_{1:n}$, the decoder then generates an output sequence $\mathbf{y}_{1:m}$ of symbols one element at a time. At each step the model consumes the previously generated symbols as additional input when generating the next. The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder, shown in the left and right halves of Figure 2.9, respectively.

The encoder is composed of a stack of N identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. A residual connection [132] is employed around each of the two sub-layers, followed by layer normalization [7]. The output of each sub-layer is $LayerNorm(x + Sublayer(x))$, where $Sublayer(x)$ is the function implemented by the sub-layer itself. To facilitate these residual connections, all sub-layers in the model, as well as the embedding layers, produce outputs of a fixed dimension d_{model}. The decoder is also composed of a stack of N identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the

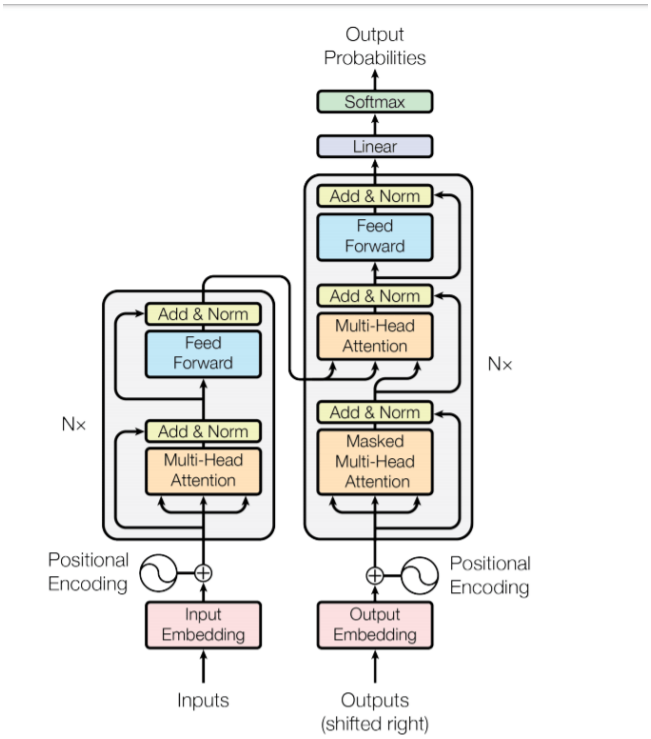


Figure 1: The Transformer - model architecture.

FIGURE 2.9: Transformer model architecture

encoder stack. Similar to the encoder, residual connections are employed around each of the sub-layers. The residual connections are followed by layer normalization. The self-attention sub-layer in the decoder stack is modified to prevent positions from attending to subsequent positions. This masking, combined with fact that the output embeddings are offset by one position, ensures that the predictions for position i can depend only on the known outputs at positions less than i. The single components of this architecture are discussed in the following subsections.

Attention function

An attention function can be viewed as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. Transformers architecture use a particular attention function: the "Scaled Dot-Product Attention" 2.10. In this case the input consists of queries and keys of dimension d_k, and values of dimension d_v. The dot products of the query is computed with all keys, divide each by $\sqrt{d_k}$, and a softmax function is applied to obtain the weights on the values. In practice, the attention function is computed on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V. We compute the matrix of outputs as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.12}$$

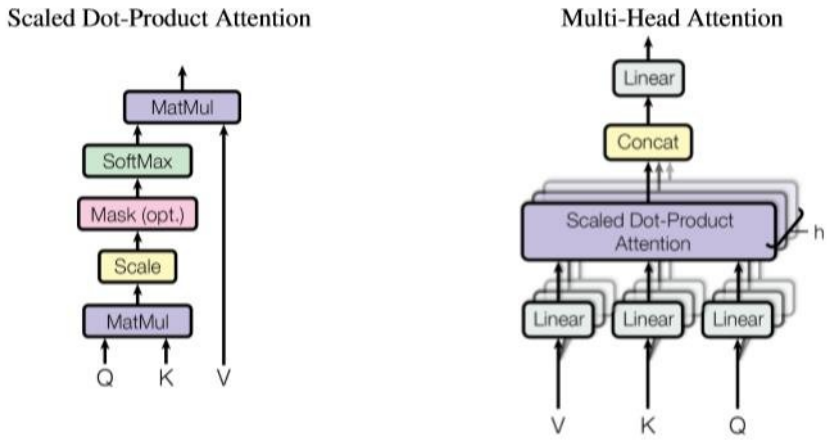


FIGURE 2.10: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

Multi-Head Attention

Instead of performing a single attention function with d_{model}-dimensional keys, values and queries, has been found beneficial to linearly project the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively. On each of these projected versions of queries, keys and values the attention function is performed in parallel, yielding d_v-dimensional output values. These are concatenated and once again projected, resulting in the final values, as depicted in Figure 2.10. Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions.

$$\begin{aligned} MultiHead(Q,K,V) &= Concat(head_1,...,head_h)W^O \\ \text{where } head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.13}$$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.

The Transformer uses multi-head attention in three different ways:

- The encoder internally contains self-attention layers. In a self-attention layer, all of the keys, values and queries come from the same place, in this case, the output of the previous layer of the encoder. The input to the multi-head self-attention is the input sequence itself (the keys, values and also the queries in various linear transformed heads)

- In the encoder-decoder attention layers, the queries come from the previous decoder layer, and the keys and values come from the output of the encoder. This allows every position in the decoder to attend over all the positions in the input sequence (similar to the typical encoder-decoder architecture)

- Similarly, self-attention layers in the decoder will allow each position to attend to all positions up to and including that position.
- To prevent the leftward information flow in the decoder, masking support is implemented inside of the scaled dot-product attention by masking out all values in the input of the softmax of the multi-head attention which corresponds to illegal connections (masking of future/subsequent words).

Position-wise Feed-Forward Networks

In addition to attention sub-layers, each of the layers in our encoder and decoder contains a fully connected feed-forward network, which is applied to each position separately and identically. This consists of two linear transformations with a ReLU activation in between.

$$FFN(\mathbf{x}) = max(0, \mathbf{x}W_1 + b_1)W_2 + b_2 \tag{2.14}$$

While the linear transformations are the same across different positions, they use different parameters from layer to layer.

Embeddings and Softmax

Similarly to other sequence transduction models, learned embeddings are used to convert the input tokens and output tokens to vectors of dimension d_{model}. A learned linear transformation and softmax function are also to convert the decoder output to predicted next-token probabilities.

Positional Encoding

Since the Transformer model contains no recurrence and no convolution, in order for the model to make use of the order of the sequence, some information about the relative or absolute position of the tokens in the sequence must be injected. To this end, "positional encodings" are added to the input embeddings at the bottoms of the encoder and decoder stacks. The positional encodings have the same dimension d_{model} as the embeddings so that the two can be summed. The transformer model uses sine and cosine functions of different frequencies:

$$\begin{aligned} PE_{(pos,2i)} &= sin(pos/10000^{2i/d_{model}}) \\ PE_{(pos,2i+1)} &= cos(pos/10000^{2i/d_{model}}) \end{aligned} \tag{2.15}$$

where pos is the position and i is the dimension. Each dimension of the positional encoding corresponds to a sinusoid. The wavelengths form a geometric progression from 2π to $10000 \cdot 2\pi$. This function should allow the model to easily learn to attend by relative positions since for any fixed offset k, PE_{pos+k} can be represented as a linear function of PE_{pos}.

2.8.1 Transformer Language Modelling

After [297] publication, several works have been focused on exploiting versions of the Transformer architecture to build language models [137, 234, 86, 243]. Historically one of the biggest issues in NLP is the shortage of training data. In NLP many distinct tasks needs to be tackled, then most task-specific datasets contain only a few thousand or in lucky cases a few hundred thousand human-labeled training examples. To help close this gap in data, those language model can be trained using the enormous amount of unannotated text available on the web. The language model can then be fine-tuned on small-data NLP tasks, resulting in substantial accuracy improvements compared to training on these datasets from scratch. This approach is called *transfer learning* and has been widely adopted recently in the NLP practice. The language model training on unlabaled corpus is usually called pretraining. In subsection 2.6.1 we mentioned a limit of word embeddings: the lack of contextuality. The output of transformers language models is practically an embedding that represents a word depending on its context: a contextual word embedding.

One of the most popular transformer-based language models is BERT [86]. Unlike these previous models, BERT is designed to pretrain deep bidirectional representations by jointly conditioning on both left and right context in all layers. This is particularly useful for example as the meaning of the words depends on both left and right side context. Consider again the word *"bank"* in the following context *"bank account"* and *"bank of the river"*, in this case, the right context is fundamental to understand the meaning of the word. However, it is not possible to train bidirectional models by simply conditioning each word on its previous and next words, since this would allow the word that's being predicted to indirectly "see itself" in a multi-layer model. To solve this problem, BERT uses the straightforward technique of masking out some of the words in the input and then condition each word bidirectionally to predict the masked words. For example an input sentence could be: *"The man went to the* $[MASK]_1$ *. He bought a* $[MASK]_2$ *of milk."*, in this case the model goal is to predict *"*$[MASK]_1 = store$*"* and *"*$[MASK]_2 = gallon$*"*. BERT also learns to model relationships between sentences by learning to predict if two sentences are consequent or not.

2.8.2 Current Trends: Efficient Transformer

Transformer model architectures became very popular in NLP due to their effectiveness across a wide range of tasks. Although they are characterized by quadratic complexity with respect to the input sequence length and the most effective Transformer model are made of a lot of layers, so the process of pretraining have a huge computational cost. Also, those models are memory heavy and slow in prediction. This represents a limitation for the wide adoption of those models. Recently, a large number of "X-former" models have been proposed: Reformer[158], Linformer[300], Performer[52], Longformer[22], to name a few. Those models attempt to improve upon the original Transformer architecture, many of them by make improvements around computational and memory efficiency. [290] provide a survey about this topic.

Chapter 3

Natural Language Generation: a brief overview

In this chapter, an overview of NLG is provided, with a specific focus on neural techniques and evaluation. Part of this chapter is inspired by [105], who provides a survey of the state of the art of NLG, however more recent advancements have been reported and additional details about evaluation in NLG are inspired by [45, 267].

NLG has been traditional divided into two major areas: text-to-text generation and data-to-text generation. The Text-to-text generation goal is to automatically produce a new coherent text as output, taking existing texts as input. Example applications that generate new texts are:

- machine translation, from one language to another e.g., [142, 228]
- fusion and summarization of related sentences or texts to make them more concise e.g., [60]
- simplification of complex texts, for example, to make them more accessible for low-literacy readers e.g., [278] or for children [197];
- automatic spelling, grammar and text correction e.g., [165, 73];
- automatic generation of peer reviews for scientific papers [16];
- generation of paraphrases of input sentences e.g., [10, 150];
- automatic generation of questions, for educational and other purposes e.g., [33, 264].

We refer to data-to-text generation when the generated texts are not grounded in existing ones but in some other kind of data. Data-to-text applications had a considerable impact in the fields of journalism and media studies [296, 62, 310]. Many applications have been developed over the years such as:

- football reports e.g., [291, 49];
- virtual 'newspapers' from sensor data [217] and news reports on current affairs [175];

- text addressing environmental concerns, such as wildlife tracking [236], personalized environmental information [301], and enhancing the engagement of citizen scientists via generated feedback [299];
- weather and financial reports [114, 255, 294, 245, 235];
- summaries of patient information in clinical contexts [141, 238, 106, 9];
- interactive information about cultural artefacts, for example in a museum context e.g., [227, 285];
- automatic image captioning [13].

Traditionally, the NLG problem of converting input data into output text was addressed by splitting it up into a number of sub-problems. The following six are frequently found in many NLG systems [252]:

- content determination: Deciding which information to include in the text under construction,
- text structuring: Determining in which order information will be presented in the text,
- sentence aggregation: Deciding which information to present in individual sentences,
- lexicalisation: Finding the right words and phrases to express information,
- referring expression generation: Selecting the words and phrases to identify the domain,
- linguistic realisation: Combining all words and phrases into well-formed sentences.

Broadly speaking, we can distinguish between three dominant approaches to NLG architectures:

- Modular architectures: by design, such architectures involve fairly crisp divisions among sub-tasks, though with significant variations among them;
- Planning perspectives: viewing text generation as planning links it to a long tradition in AI and affords a more integrated, less modular perspective on the various sub-tasks of NLG;
- Integrated or global approaches: now the dominant trend in NLG (as it is in NLP more generally), such approaches cut across task divisions, usually by placing a heavy reliance on statistical learning of correspondences between inputs and outputs.

My research will be focused on the global approach, nowadays more attractive thanks to the growing capabilities of deep learning techniques.

3.1 Language Models for Natural Language Generation

As mentioned in section 2.5 any language model can be used for generating texts: after training a language model on a given collection of text, one can generate random sentences from the model according to their probability using the following process: predict a probability distribution over the first word conditioned on the start symbol and draw a random word according to the predicted distribution. Then, predict a probability distribution over the second word conditioned on the first, and so on, until predicting the end-of-sequence symbol. When generating a sentence from a trained language model in this way, one can either choose the highest scoring word at each step or sample a random word according to the predicted distribution. This approach is called a greedy search, another option is to use beam search in order to find a sequence with a globally high probability. Instead of greedily choosing the most likely next step as the sequence is constructed, the beam search expands all possible next steps and keeps the k most likely, where k is a user-specified parameter and controls the number of beams or parallel searches through the sequence of probabilities.

3.2 Neural Natural Language Generation

As in NLP in general neural networks, methods are currently the most popular in the NLG community. NNs have scored notable successes in language modelling using FFNs [26, 271] and RNNs. RNNs main advantage over standard language models is that they handle sequences of varying lengths while avoiding both data sparseness and an explosion in the number of parameters through the projection of histories into a low-dimensional space so that similar histories share representations. The sequence to sequence architecture is widely adopted in NLG: this decoupling between encoding and decoding makes it possible in principle to share the encoding vector across multiple NLP tasks in a multi-task learning setting [89, 195]. Those architecture suites well for text-to-text tasks such as Machine Translation, which can be thought of as requiring the mapping of variable-length input sequences in the source language, to variable-length sequences in the target [148, 53]. It is easy to adapt this view to data-to-text NLG. For example, [94] adapt the sequence to sequence models for generating text from abstract meaning representations (AMRS). Further important development within the sequence to sequence paradigm is the use of attention-based mechanisms, which obviates the need for direct input-output alignment. In NLG, many approaches to response generation in an interactive context (such as dialogue or social media posts) adopt this architecture. For example, [304] use semantically-conditioned LSTMs to generate the next act in a dialogue; a related approach is taken by [282], who use RNNs to encode both the input utterance and the dialogue context, with a decoder to predict the next word in the response. A popular architecture that has been widely adopted for abstractive summarization is the pointer network [272]: this architecture augments the standard sequence to sequence models by using a hybrid pointer-generator network that can copy words from the source text via pointing, which aids accurate reproduction of information while retaining the ability to produce novel words through the generator. The same work also introduces the coverage attention to keep

track of what has been summarized and discourages repetition, very frequent in sequence to sequence models.

3.2.1 Transformer based NLG

Transformer based models improved the state of the art in language modelling, for this reason, they have been widely adopted also for NLG.

A popular Transformer architecture in NLG is the decoder-only architecture [184], in which the Transformer encoder is dropped. [184] used decoder-only architecture to train a summarization system, while GPT [242] and GPT-2 [243] used the same approach to train language models. The largest version of GPT-2 is a 1.5B parameter model, which thanks to its size can generate impressively natural sounding texts. An even bigger version called GPT-3 [35] has recently attracted a lot of media attention.

Another popular model that has been widely adopted in NLG is BART [176]. This model is a denoising autoencoder: is trained by providing text corrupted with an arbitrary noising function as input, and the original text as a learning object. It uses a Transformer based neural architecture which can be seen as generalizing BERT bidirectional encoder and simultaneously GPT left-to-right decoder. This model obtained impressive results in the abstractive summarization task.

A few transformer-based architecture that tackle also data-to-text tasks has been proposed. VideoBERT [286] is a joint visual-linguistic model to learn high-level features without any explicit supervision and it has been used for video captioning. [117] introduced a Transformer-based data-to-text generation model which learns content selection and summary generation in an end-to-end fashion, this model has been applied for NBA Basketball matches summaries generation, exploiting the dataset provided by [307].

3.3 Control Variation in NLG

NLG is focused on generating texts that deliver the expected meaning, however, there is more than one way of saying the same thing. The control of stylistic and affect variation is an important aspect of NLG systems. Several early non-neural contributions in this area have been proposed, [105] provides an extensive overview of those techniques. Here we will focus on neural techniques.

A number of models focus on response generation, where the task is to generate a response, given an utterance. Thus, these models fit well within the sequence to sequence. Often, these models exploit social media data, especially from Twitter. For example, [177] proposed a persona-based model in which the LSTM decoder is conditioned on embeddings obtained from tweets pertaining to individual authors. An alternative model conditions on both speaker and addressee profiles, with a view to incorporating not only the ‘persona’ of the generator but its variability with respect to different interlocutors. [133], also working on Twitter data, condition their decoder on personality features extracted from tweets based on the ‘Big Five’ model [28], rather than on speaker-specific embeddings. This has the advantage of not enabling the generator to be tuned to specific personality settings, without re-training to adapt

to a particular speaker style. While their personality-based model does not beat the model introduced in [177], a human evaluation showed that judges were able to identify high-trait responses as more expressive than low-trait responses, suggesting that the conditioning was having a noticeable impact on style. In a dialogue context, [4] proposed to achieve affective responses on three levels: (a) by augmenting word embeddings with data from an affective dictionary; (b) by decoding with an affect-sensitive beam search; and (c) by training with an affect-sensitive loss function.

On the other hand, a number of models condition an LSTM on attributes reflecting affective or personality traits, with a view to generating strings that express such traits. [108] used LSTMs trained on speech corpora conditioned on affect category and emotional intensity to drive lexical choice. [138] used variational auto-encoders [155, 257] to control the stylistic parameters of generated texts individually. They experimented on controlling sentiment and tense, but restricted the generation to sentences of up to 16 words. By contrast, [95] extend the range of parameters used to condition the LSTM, with two content-related attributes (sentiment and theme) and four stylistic parameters (length, whether the text is descriptive, whether it has a personal voice, and whether the style is professional). Their generator is trained on a corpus of movie reviews. Similarly, [90] propose an attribute-to-sequence model for product review generation based on a corpus of Amazon user reviews. The conditioning includes the reviewer id, reminiscent of the persona-based response model of [177], however, they also include the rating, which functions to modulate the affect in the output. Their model incorporates an attentional mechanism to concentrate on different parts of the input encoding when predicting the next word during decoding.

Large transformer-based language models trained on huge text corpora have shown unparalleled generation capabilities, however, controlling variation is non-trivial. [151] proposed a language model with style conditioning that relies on control codes obtained from metadata of the training data. [319] exploited reinforcement learning to control GPT-2 text generation. [74] proposed to combine a pre-trained language model with one or more simple attribute classifiers that guide text generation without any further training of the language model. For each token, the mean hidden representation of all tokens so far is fed into a style classifier. A backward pass through the classifier and generator is performed, and the gradients are used to update the activations in the generator's attention layers. These forward and backward passes are repeated several times per time step, and the following token is then sampled. [162] used a similar approach but improved greatly the efficiency of the generator. [287] argued that the original unconditioned language models are sufficient for conditioned NLG and they proposed four different techniques to condition GPT-2 language model without any additional training.

3.4 NLG systems evaluation

NLG evaluation is challenging and marked by a great deal of variety [105] mainly because many NLG tasks are open-ended. For example, a dialogue system can generate multiple plausible responses for the same user input. A document can be summarized in different

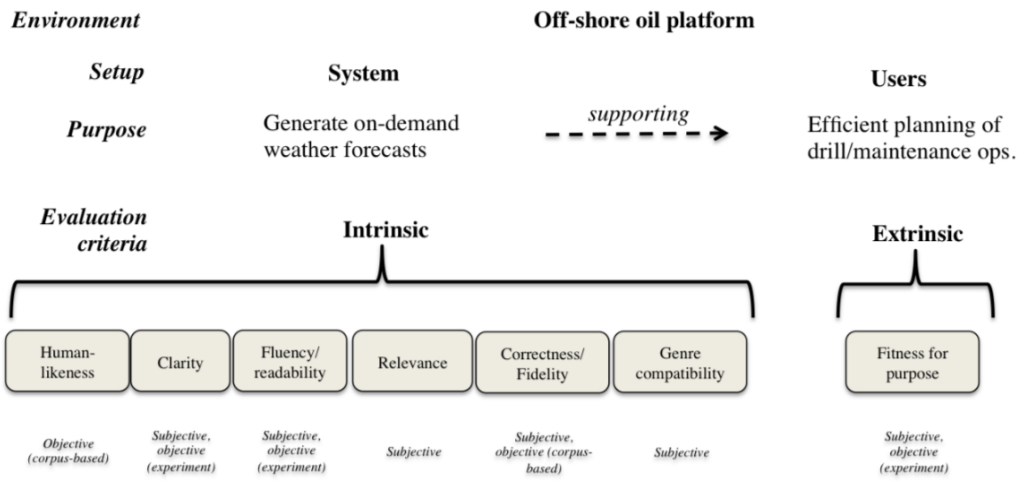


FIGURE 3.1: Source: [105]. Hypothetical evaluation scenario: a weather report generation system embedded in an offshore oil platform environment. Possible evaluation methods focusing on different questions are highlighted at the bottom, together with the typical methodological orientation (subjective/objective) adopted to address them.

ways. Moreover, NLG system is characterized by variable input. Therefore, human evaluation remains the gold standard for almost all NLG tasks. However, human evaluation is expensive, and researchers often resort to automatic metrics for quantifying day-to-day progress and for performing automatic system optimization [45]. [105] highlights some topical issues in NLG evaluation. [45, 267] provide two extensive surveys on NLG evaluation covering an extensive range of evaluation techniques. In this section, we will summarize the most salient issues and the evaluation techniques relevant for the scope of this thesis by relying on the cited works.

By way of an overview of these issues, consider the hypothetical scenario sketched in 3.1, which is loosely inspired by work on various weather-reporting systems developed in the field. This NLG system is embedded in the environment of an offshore oil-rig; the relevant features of the setup (in the sense of [147]) are the system itself and its users, here a group of engineers. While the task of the system is to generate weather reports from numerical weather prediction data, its ultimate purpose is to facilitate users' planning of drilling and maintenance operations. Figure 3.1 highlights some of the common questions addressed in NLG evaluation, together with a broad typology of the methods used to address them, in particular, whether they are objective – that is measurable against an external criterion, such as corpus similarity or experimentally obtained behavioural data – or subjective, requiring human judgements. A fundamental methodological distinction, due to [147], is between intrinsic and extrinsic evaluation methods. In the case of NLG, an intrinsic evaluation measures the performance of a system without reference to other aspects of the setup, such as the system's effectiveness in relation to its users. In the example scenario, questions related to text quality, the correctness of output and readability qualify as intrinsic, whereas the question of whether the system actually achieves its goal in supporting adequate decision-making on the offshore platform is extrinsic.

In order to evaluate all these different aspects, it is possible to exploit a wide range of different techniques, which are grouped into three main categories:

- Human Evaluation. The most natural way to evaluate the quality of a text generator is to involve humans as judges. Naive or expert subjects are asked to rate or compare texts generated by different NLG systems or to distinguish machine-generated texts from human-generated texts. Most human evaluations are task-specific, and thus need to be designed and implemented differently for the outputs of different tasks.
- Untrained Automatic Metrics. This category, also known as automatic metrics, is the most commonly used in the research community. These evaluation methods compare machine-generated texts to human-generated texts (references) from the same input data using metrics based on string overlap, content overlap, string distance, or lexical diversity, such as n-gram match and distribution similarity.
- Machine-Learned Metrics. These metrics are based on machine-learned models which can be viewed as digital judges that simulate human judges.

In the following subsections, we will dive into those three evaluation techniques categories.

3.4.1 Human Evaluation

The ultimate goal of NLG systems is to generate text that is valuable to people. For this reason, human evaluations are typically viewed as the most important form of evaluation for NLG systems and are held as the gold standard to evaluate automatic metrics. While human evaluations give the best insight into how well a model performs in a task, it is worth noting that human evaluations also pose several challenges. First, human evaluations can be expensive and time-consuming to run, especially for tasks that require extensive domain expertise. While online crowd-sourcing platforms have enabled researchers to run experiments on a larger scale at a lower cost, they come with their own problems, such as maintaining quality control [143, 216]. Furthermore, even with a large group of annotators, there are some dimensions of generated text that are not well-suited to human evaluations, such as diversity [130] or style [79]. Human evaluation can be intrinsic or extrinsic. The intrinsic evaluation is typically done by generating several samples of text from a model and asking human evaluators to score their quality along some dimensions (e.g., fluency, coherence, correctness, etc.). The simplest way to get this type of evaluation is to show the evaluators the generated texts one at a time and have them judge their quality individually. They are asked to vote whether the text is good or bad, or to make more fine-grained decisions by marking the quality along a Likert or sliding scale. However, judgments in this format can be inconsistent and comparing these results is not straightforward; [2] find that analysis on NLG evaluations in this format is often done incorrectly or with little justification for the chosen methods. To more directly compare a model's output against baselines, model variants, or human-generated text, intrinsic evaluations can also be performed by having people choose which of two generated texts they prefer, or more generally, rank a set of generated texts. This

comparative approach has been found to produce higher inter-annotator agreement [43] in some cases. However, while it captures models' relative quality, it does not give a sense of the absolute quality of the generated text. One way to address this is to use a method like RankME [225], which adds magnitude estimation [12] to the ranking task, asking evaluators to indicate how much better their chosen text is over the alternatives. Comparison based approaches can become prohibitively costly (by requiring lots of head-to-head comparisons) or complex (by requiring participants to rank long lists of output) when there are many models to compare, though there are methods to help in these cases. For example, best-worst scaling [190] has been used in NLG tasks [156, 160] to simplify comparative evaluations; best-worst scaling asks participants to choose the best and worst elements from a set of candidates, a simpler task than fully ranking the set that still provides reliable results.

Almost all the text generation tasks today are evaluated with intrinsic human evaluations. Machine translation is one of the text generation tasks in which intrinsic human evaluations have made a huge impact. Two of the most frequent aspects took into consideration are adequacy (in the respect to the source) and fluency. Fluency is also evaluated in several text generation tasks such as document summarization [46] and image captioning [170]. While fluency and adequacy have become standard dimensions of human evaluation for machine translation, not all text generation tasks have an established set of dimensions that researchers use. Nevertheless, there are several dimensions that are common in human evaluations for generated text. As with adequacy, many of these dimensions focus on the contents of the generated text. Factuality is important in tasks that require the generated text to accurately reflect facts described in the context. For example, in tasks like data-to-text generation or summarization, the information in the output should not contradict the information in the input data table or news article. This is a challenge to many neural NLG models, which are known to "hallucinate" information [136, 303]. Even if there is no explicit set of facts to adhere to, researchers may want to know how well the generated text follows rules of commonsense or how logical it is. For generation tasks that involve extending a text, researchers may ask evaluators to gauge the coherence or consistency of a text. Other dimensions focus not on what the generated text is saying, but how it is being said. As with fluency, these dimensions can often be evaluated without showing evaluators any context. This can be something as basic as checking for simple language errors by asking evaluators to rate how grammatical the generated text is. It can also involve asking about the overall style, formality, or tone of the generated text, which is particularly important in style-transfer tasks.

Extrinsic evaluations are the most meaningful evaluation as they show how a system actually performs in a downstream task, but they can also be expensive and difficult to run. For this reason, intrinsic evaluations are more common than extrinsic evaluations. Extrinsic evaluation can be measured from two different perspectives: a user's success in a task and the system's success in fulfilling its purpose [131]. For example, [311] evaluate from the user perspective by measuring the number of mistakes subjects made when they followed automatically generated instructions. From a system perspective [253] generate personalized smoking cessation letters and report how many recipients actually gave up smoking. Extrinsic human evaluations are commonly used in evaluating the performance of dialog [83] and have

made an impact on the development of dialogue modelling systems. Various approaches have been used to measure the system's performance when talking to people, such as measuring the conversation length or asking people to rate the system. The feedback is collected by real users of the dialogue system [29, 168, 317] at the end of the conversation.

Not all NLG evaluation tasks can be performed by any subset of speakers of a given language. Specialized groups of evaluators can be useful when testing a system for a particular set of users, as in extrinsic evaluation settings. Some other specific cases may require specialised skill-sets. However, for many NLG evaluation tasks, no specific expertise is required of the evaluators other than a proficiency in the language of the generated text. In this cases evaluations performed either in-person or online. The benefits of in-person evaluation are that it is easier to train and interact with participants and that it is easier to get detailed feedback about the study and adapt it as needed. Researchers also have more certainty and control over who is participating in their study, which is especially important when trying to work with a more targeted set of evaluators. However, in-person studies can also be expensive and time-consuming to run. For these reasons, in-person evaluations tend to include fewer participants, and the set of people in proximity to the research group may not accurately reflect the full set of potential users of the system. In-person evaluations may also be more susceptible to response biases, adjusting their decisions to match what they believe to be the researchers' preferences or expectations [222, 229]. To mitigate some of the drawbacks of in-person studies, online evaluations of generated texts have become increasingly popular. While researchers could independently recruit participants online to work on their tasks, it is common to use crowdsourcing platforms that have their own users whom researchers can recruit to participate in their task, either by paying them a fee. These platforms allow researchers to perform large-scale evaluations in a time-efficient manner, and they are usually less expensive to run. They also allow researchers to reach a wider range of evaluators than they would be able to recruit in-person. However, maintaining quality control online can be an issue, and the demographics of the evaluators may be heavily skewed depending on the user base of the platform [87, 249]. Furthermore, there may be a disconnect between what evaluators online being paid to complete a task would want out of an NLG system and what the people who would be using the end product would want.

Evaluating generated natural language will always include some degree of subjectivity. Evaluators may disagree in their ratings, and the level of disagreement can be a useful measure to researchers. High levels of inter-evaluator agreement generally mean that the task is well-defined and the differences in the generated text are consistently noticeable to evaluators, while low agreement can indicate a poorly defined task or that there are no reliable differences in the generated text. The agreement is usually low in generated text evaluation tasks, lower than what is typically considered "acceptable" on most agreement scales (Amidei et al., 2018, 2019a). However, as Amidei et al. (2018) point out, given the richness and variety of natural language, pushing for the highest possible inter-annotator agreement may not be the right choice when it comes to NLG evaluation. Different measure to compute the are available, for example, percent agreement, Cohen's κ [63], Fleiss' κ [99] and Krippendorff's α [164]

3.4.2 Untrained Automatic Metrics

Human evaluation is costly and time-consuming to design and run and the results are not always repeatable [24]. Thus, automatic evaluation metrics are employed as an alternative in both developing new models and comparing them against the state-of-the-art. Untrained automatic metrics for NLG compute a score that indicates the similarity between an automatically generated text and human written reference text. Untrained automatic evaluation metrics are fast and efficient and are widely used to quantify the day-to-day progress of model development. [45] group the untrained automatic evaluation methods in five categories:

- n-gram overlap metrics
- distance-based metrics
- diversity metrics
- content overlap metrics
- grammatical feature-based metrics

n-gram overlap metrics

n-gram overlap metrics are commonly used for evaluating NLG systems and measure the degree of "matching" between machine-generated and reference human-authored texts. Follows a non-comprehensive review of the most popular n-gram overlap metrics. The most famous metric is BLEU [231] and it is used to measure the similarity between two sentences. Originally proposed for machine translation, it compares a candidate text to one or more reference. BLEU is a weighted geometric mean of n-gram precision scores. Text generation research, especially when focused on short text generation like sentence-based machine translation or question generation, has successfully used BLEU for benchmark analysis with models since it is fast, easy to calculate, and enables a comparison with other models on the same task. However, BLEU has some drawbacks for NLG tasks where contextual understanding and reasoning is the key: it considers neither semantic meaning nor sentence structure. It does not handle morphologically rich languages well, nor does it map well to human judgments [289].

ROUGE [181] is a popular set of metrics for evaluating automatic summarization of long texts consisting of multiple sentences or paragraphs. Although mainly designed for evaluating single- or multi-document summarization, it has also been used for evaluating short text generation. ROUGE includes a large number of distinct variants, including eight different n-gram counting methods to measure n-gram overlap between the generated and the ground-truth (human-written) text. ROUGE also includes a setting for word-stemming of summaries and an option to remove or retain stop-words. Compared to BLEU, ROUGE focuses on recall rather than precision. Additionally, ROUGE includes the mean or median score from individual output text, which allows for a significance test of differences in system-level ROUGE scores, while this is restricted in BLEU [121]. However ROUGE evaluates the adequacy of the generated output text by counting how many n-grams in the generated output

text matches the n-grams in the reference text. This is considered a bottleneck of this measure, especially for long-text generation tasks [152], because it doesn't provide information about the narrative flow, grammar, or topical flow of the generated text, nor does it evaluate the factual correctness of the summary compared to the corpus it is generated from.

The Metric for Evaluation of Translation with Explicit ORdering (METEOR) [172] is a metric designed to address some of the issues found in BLEU and has been widely used for evaluating machine translation models and other NLG models. Compared to BLEU, which only measures the precision, METEOR is based on the harmonic mean of the unigram precision and recall, in which recall is weighted higher than precision. METEOR has several variants that extend exact word matching that most of the metrics in this category do not include, such as stemming and synonym matching. These variants address the problem of reference translation variability, allowing for morphological variants and synonyms to be recognized as valid translations. The metric has been found to produce a good correlation with human judgments at the sentence or segment level [172].

Distance-based metrics

A distance-based metric in NLG applications uses a distance function to measure the similarity between two text units. Edit distance, one of the most commonly used evaluation metrics in natural language processing, measures how dissimilar two text units are based on the minimum number of operations required to transform one text into the other. Examples of Edit distance-based metrics are Word error rate (WER), minimum edit distance (MED) and Translation edit rate (TER) [280]. In addition to n-gram-based similarity metrics embedding-based similarity measures are commonly used. Even though the embedding vectors are learned using supervised or unsupervised neural network models, some vector-similarity metrics assume the embeddings are pre-trained and simply used as input to calculate the metric. Some examples are MEANT [186], YISI [187], Word Mover's Distance (WMD) [166], Sentence Mover's Distance (SMD) [59] and Frechet Inception Distance (FID) [134].

Diversity metrics

The lexical diversity score measures the breadth and variety of word usage in writing. A few metrics designed to measure the quality of the generated text in terms of lexical diversity that are available. Type-Token Ratio (TTR) is a measure of lexical diversity [258], mostly used in linguistics to determine the richness of a writer's or speaker's vocabulary. It is computed as the number of unique words (types) divided by the total number of tokens in a given segment of language. Although intuitive and easy to use, TTR has a major problem: it is sensitive to text length. [318] propose SELF-BLEU as a diversity evaluation metric by measuring the differences between generated sentences and references or other generated texts. In a sense, it is the opposite of BLEU, which assesses how similar two sentences are. Taking a generated sentence to be evaluated as the hypothesis and the other sentences as references, SELF-BLEU calculates a BLEU score for every generated sentence and defines the average of these BLEU scores as the SELF-BLEU score of the to-be-evaluated text. A lower SELF-BLEU score

implies higher diversity. To overcome the length sensitiveness of TTR [204] proposed HD-D a hypergeometric distribution function.

Content overlap metrics

Semantic content matching metrics define the similarity between human-written and model generated text by extracting explicit semantic information units from text beyond n-grams. These metrics operate on semantic and conceptual levels and are shown to correlate well with human judgments. An example of this kind of metric used for summarization is PYRAMID [221], however this metric is not fully automatic, PEAK: Pyramid Evaluation via Automated Knowledge Extraction [309] is presented as a fully automated variant of the PYRAMID model. SPICE [3] is a content overlap metrics designed for image captioning. Other text generation work has used the confidence scores obtained from semantic similarity methods as an evaluation metric. Examples of semantic similarity methods used for this goal are Semantic Textual Similarity (STS) [1], Paraphrase identification (PI) [17, 88], Textual entailment (TE) [71, 32], Machine Comprehension (MC) [244].

Grammatical feature-based metrics

Grammatical feature-based metrics capture the similarity between a reference and a hypothesis text at a structural level to capture the overall grammatical or sentence structure similarity. POS tags have been commonly used in machine translation evaluation [72, 237, 128]. Always in machine translation, several works have enriched their evaluation criteria by leveraging syntactic analysis [183, 188, 312].

3.4.3 Machine-Learned Metrics

Almost all the most popular untrained evaluation metrics assume that the generated text has significant word (or n-gram) overlap with the ground-truth text. However, this assumption does not hold for many NLG tasks, such as a social chatbot, which permit significant diversity and allow multiple plausible outputs for a given input. One solution to this problem is to use embedding-based metrics, which measure semantic similarity rather than word overlap, but those methods cannot help in situations where the generated output is semantically different from the reference. In these cases, we can build machine-learned models trained on ground truth data.

Neural approaches to sentence representation learning seek to capture semantic and syntactic meanings of sentences from different perspectives and topics and to map a sentence onto an embedding vector using DNN models. As with word embeddings, NLG models can be evaluated by embedding each sentence in the generated and reference texts. Several techniques to encode sentences are available e.g. [139, 157, 189, 69]. Also, Transformer language models such as BERT use contextualized word embeddings to represent sentences. Even though these PLMs outperform the earlier models such as DSSMs, they are more computationally expensive to use.

[315] proposed a way to tackle the problem of factual correctness in summarization models. Focusing on summarizing radiology reports, they extend pointer networks for abstractive summarization by introducing a reward-based optimization that trains the generators to obtain more rewards when they generate summaries that are factually aligned with the original document. Specifically, they design a fact extractor module so that the factual accuracy of a generated summary can be measured and directly optimized as a reward using policy gradient. This fact extractor is based on an information extraction module and extracts and represents the facts from generated and reference summaries in a structured format. The summarization model is updated via reinforcement learning using a combination of the negative log-likelihood loss, a ROUGE-based loss, and a factual correctness-based loss. Their work suggests that for domains in which generating factually correct text is crucial, a carefully implemented information extraction system can be used to improve the factual correctness of neural summarization models via reinforcement learning. To address the same goal [92] introduced a question-answering-based parametric evaluation model named Answering Performance for Evaluation of Summaries (APES). Their evaluation model is designed to evaluate document summarization and is based on the hypothesis that the quality of a generated summary is associated with the number of questions from a set of relevant ones that can be answered by reading the summary. To build such an evaluator to assess the quality of generated summaries, they introduce two components: (a) a set of relevant questions for each source document and (b) a question-answering system. Thus, for each generated summary, metrics can be derived based on the accuracy of the question answering system in retrieving the correct answers from each of the associated triplets.

For more creative and open-ended text generation tasks, such as chit-chat dialogue, story generation, or online review generation, current evaluation methods are only useful to some degree. As we mentioned at the beginning of this section, word-overlap metrics are ineffective as there are often many plausible references in these scenarios and collecting all is impossible. Even though human evaluation methods are useful in these scenarios for evaluating aspects like coherency, naturalness, or fluency, aspects like diversity or creativity may be difficult for human judges to assess as they have no knowledge about the dataset that the model is trained on. Also, a common issue is that conducting a human evaluation for every new generation task can be expensive and not easily generalizable. To calibrate human judgments and automatic evaluation metrics, model-based approaches that use human judgments as attributes or labels have been proposed. [191] formulated automatic dialogue evaluation as a learning problem. They present an evaluation model (ADEM) that learns to predict human-like scores to input responses, using a new dataset of human response scores. With the motivation that a good evaluation metric should capture both the quality and the diversity of the generated text, [130] propose a new evaluation metric named Human Unified with Statistical Evaluation (HUSE), which focuses on more creative and open-ended text generation tasks, such as dialogue and story generation. Different from the ADEM metric, which relies on human judgments for training the model, HUSE combines statistical evaluation and human evaluation metrics in one model.

Given the strong performance of BERT across many tasks, there has been work that uses

BERT or similar pre-trained language models for evaluating NLG tasks, such as summarization and dialogue response generation. Here, we summarize some of the recent work that fine-tunes BERT to use as evaluation metrics for downstream text generation tasks. One of the BERT-based models for semantic evaluation is BERTSCORE [316]: it leverages the pre-trained contextual embeddings from BERT and matches words in candidate and reference sentences by cosine similarity. It has been shown to correlate well with human judgments on sentence-level and system-level evaluations. [149] present a new BERT-based evaluation method called ROBERTA-STS to detect sentences that are logically contradictory or unrelated, regardless of whether they are grammatically plausible. Using ROBERTA [185] as a pre-trained language model, ROBERTA-STS is fine-tuned on the STS-B dataset to learn the similarity of sentence pairs on a Likert scale. Another evaluation model is fine-tuned on the Multi-Genre Natural Language Inference Corpus in a similar way to learn to predict logical inference of one sentence given the other. Both model-based evaluators have been shown to be more robust and correlate better with human evaluation than automatic evaluation metrics such as BLEU and ROUGE. Another recent BERT-based machine-learned evaluation metric is BLEURT [273], which is proposed to evaluate various NLG systems. The evaluation model is trained as follows: a checkpoint from BERT is taken and fine-tuned on synthetically generated sentence pairs using automatic evaluation scores such as BLEU or ROUGE, and then further fine-tuned on system-generated outputs and human-written references using human ratings and automatic metrics as labels. The fine-tuning of BLEURT on synthetic pairs is an important step because it improves the robustness to quality drifts of generation systems.

The quality of many NLG models can be evaluated for multiple aspects, such as adequacy, fluency, and diversity. For this reason composite metrics have been proposed e.g. [179, 277].

Chapter 4

Language Complexity

4.1 Introduction

Linguistic complexity is a well-studied and multifaceted variation aspect for which several measures have been proposed in different frameworks ranging from First and Second Language Acquisition, language typology and readability assessment. Such measures depend on the perspective from which linguistic complexity is considered. According to one established distinction, linguistic complexity should be divided into an *absolute* vs a *relative* notion [207]. While the former is driven by theory and aims at assessing the complexity of a language according to some formal properties of the linguistic system, the latter defines complexity in relation to the language user (e.g. speaker, listener or learner) thus considering the complexity in terms of processing difficulty. From this second perspective, sentence complexity is analyzed in terms of cognitive load, which can be inferred using both off-line (e.g. complexity judgments, error rates on a comprehension test, preference for a structure over a meaning-equivalent one in elicited production tasks) and online processing measures (e.g. eye-tracking data such as total gaze time, fixation duration and pupil dilation). To operationalize factors underlying sentence processing performance, several complexity metrics have been proposed which consider properties of single word and sentence, as well as experience-based expectations. Word-level predictors shown to correlate with greater processing difficulties are e.g. word frequency, age of acquisition, root frequency effect, orthographic neighbourhood frequency. At the syntactic level, a well-studied measure of sentence complexity takes into account dependency length [109, 110], which has been used to explain a wide range of psycholinguistic phenomena, such as the subject/object relative clauses asymmetry or the garden path effect in main verb/reduced-relative ambiguities [119, 284], as well as variations in word order patterns [113], also in a diachronic perspective [124]. Alternatively, processing difficulty has been explained in terms of surprisal [125]. Computational models to calculate lexical and syntactic surprisal have been developed by e.g. [259] using a broad-coverage probabilistic PCFG parser and [82], who introduced Prediction Theory, which aims at unifying Dependency Length Theory with syntactic surprisal, by making use of a psycholinguistically-motivated version of tree-adjoining grammar.

Unlike more conventional studies on human sentence processing carried out in experimental settings, we carried out a study [37] in which we rely on crowdsourcing methods to investigate how people perceive sentence complexity. The reliability of crowdsourced data

for linguistics and computational linguistics research is well acknowledged as shown in the survey by [218] proving that the quality of findings obtained from the crowd is comparable, if not higher, to controlled laboratory experiments. In addition, crowdsourcing reaches a broader population, in terms of age, education, profession etc. and it is thus more suitable to catch the "layman" intuition of sentence complexity. For these reasons, this method has been used in recent works in the field of readability and text simplification; it is the case [171, 61, 36] where the crowd was asked to evaluate the level of complexity or the degree of informativeness of simplified sentences compared to the original one.

In our study, we adopted a similar perspective relying on a crowdsourcing approach to collect a wide resource containing multiple annotations of sentence complexity given by humans. Unlike traditional studies which typically assess either lexical or structural complexity phenomena, we focused on the analysis of a wide set of linguistic features to investigate how all contribute to human perception of sentence complexity. This choice is also motivated by previous studies focused on the "form" of a text all related to the assessment of complexity, e.g. readability assessment [65], first language acquisition [266] and Native Language Identification [199].

4.2 Our Contributions

Our contribution to the study of sentence complexity is multiple:

- we address two research questions aimed to investigate the role played by a set of linguistic phenomena in characterizing a) the agreement among annotators when they rated the sentences independently from the assigned score and b) the human perception of complexity.
- we introduce a new crowdsourcing-based method to assess how people perceive sentence complexity and we test it for two languages;
- we collect two corpora of sentences annotated by humans with a judgment of complexity;

The two research questions refer to two phenomena that are by definition highly subjective and difficult to define. Our study intends to address this vagueness providing the following main contributions: i) detecting the main linguistic phenomena involved in the prediction of agreement and ii) which phenomena characterize a sentence that is perceived as complex by a high number of human subjects.

All the data discussed here are made available at www.italianlp.it/resources/.

4.3 Approach

We collected a dataset of rated sentences through a crowdsourcing task in which annotators were asked to give a score of complexity to a sentence. The task was carried out in two

languages, Italian and English, which have different morpho-syntactic and syntactic properties such as morphological richness and word order freedom. This choice was aimed to investigate whether there are linguistic complexity parameters shared by typologically different languages. Starting from the collected rated sentences, we automatically extracted a wide set of features spanning across multiple levels of linguistic description, which have been acknowledged in the literature on human sentence processing to be involved in sentence complexity. The contribution of these features in modelling the perception of sentence complexity was tested in two different scenarios: i) a classification experiment to assess which features contribute more in the automatic prediction of the degree of agreement among annotators and which features vary in a statistically significant way between agreed and not-agreed sentences; ii) a regression experiment to evaluate if the considered features allow predicting the complexity judgment assigned by humans and how they contribute to the prediction.

In what follows, we introduce the three main ingredients of our approach, i.e. the set of linguistic features (Section 4.3.1), the datasets of sentences (Section 4.3.2) and the crowdsourcing task (Section 4.3.3). In the rest of this chapter, we describe the experimental scenarios raised by our two research questions and discuss the results (Sections 4.4 and 4.5).

4.3.1 Linguistic Features

The set of features considered in this study captures different aspects of sentence complexity.

Raw text features:

word length, i.e. average number of characters per words (*char_tok* in all tables and figures that follow) and **sentence length**, i.e. average number of words per sentence (*n_tokens*), which are typically used as a proxy of lexical and syntactic complexity in traditional readability metrics [65];

Morpho-syntactic features:

distribution of part-of-speech types; **type/token ratio**, calculated as the ratio between the number of lexical types, the number of tokens, in terms of both lemma and forms (*ttr_form*, *ttr_lemma*); **verbal features**, i.e. the distribution of verbs according to mood (*verbs_mood*), tense (*verbs_tense*) and persons (*verbs_num_per*), and **lexical density** (*lex_density*), calculated as the ratio of content words (verbs, nouns, adjectives and adverbs) to the total tokens in a text. Psycholinguistic studies highlight that higher lexical density implies a greater cognitive load [112];

Syntactic features:

probability of syntactic dependency types e.g. subject, direct object, modifier, etc., calculated as the *distribution* of each type out of the total dependency types. Some syntactic relations have been shown to be harder to process, e.g. object-relative clauses and

prepositional-phrase attachments [111, 110], or the subject and object relations especially in free word-order languages;
distribution of verbal roots, i.e. the *distribution of verbal roots* out of the total of sentence roots. A lower percentage of verbal roots implies a higher number of nominal sentences which have a less-standard structure due to verb ellipsis thus possibly causing processing ambiguity;
parse tree depth features: the *depth of the whole parse tree* (*max_depth*), calculated in terms of the longest path from the root of the dependency tree to some leaf; the *depth of embedded complement chains* governed by a nominal head and including either prepositional complements or nominal and adjectival modifiers, calculated as the *total* number of prepositional chains (*n_prep_chains*) and the *average* depth of chains (*prep_chain_l*); the *distribution of embedded complement chains by depth*, calculated as the number of chains out of the total number of chains in a sentence (*prep_depth*). All these features are related to length factors and correlate with processing difficulty [101], as in the case of long sequences of embedded prepositional complements;
verbal predicate features: the distribution of verbal head (*verb_head*); the *arity of verbs*, meant as the average number of instantiated dependency links sharing the same verbal head covering both arguments and modifiers *verb_arity*); the *distribution of verbal head by arity*, calculated as the total number of verbal heads with the same arity in a sentence (*verb_head_arity*); the *relative ordering of subject and object with respect to the verbal head* (*order_subj* and *order_obj*);
subordination features include the *distribution of main vs. subordinate clauses* (*n_subord_clauses* and *n_princ_clauses*; the *average depth of chains of embedded subordinate clauses*, calculated as the *total* number of subordinate chains (*n_subord_chain*) and the *average* depth of subordinate chains (*subord_chain_l*); the *distribution of embedded subordinate clauses chains by depth*, calculated as the number of chains out of the total number of chains in a sentence (*subord_depth*). We also calculated the order of the subordinate clause with respect to the main clause (*order_subord*), since according to e.g. [212], sentences containing subordinate clauses in post–verbal than in pre–verbal position are easier to process;
length of dependency links calculated as the number of words between the syntactic head and the dependent: the feature includes the *length of all dependency links* (*links_len)* and of the *maximum dependency links* (*max_links_l*). It is widely known that long-distance constructions cause cognitive load [109, 113];
clause length measured as the number of tokens occurring within a clause (*token_clause*). Syntactic metrics relying on this feature, such as the T-Unit [140], are widely used e.g. in first and second language acquisition to assess the development of syntactic competence.

4.3.2 Data

The experiments were carried out on a subset of sentences extracted from two manually revised treebanks. We chose this kind of data in order to prevent possible errors produced by the automatic annotation of sentences. Specifically, we considered the newspaper section of the Italian Universal Dependency Treebank (UDT) [279] and the automatically converted Wall Street Journal section of the Penn Treebank [205]. Since we wanted to investigate the human

perception of complexity with respect to standard language, we didn't use the English version of the UDT containing different genres of web media (e.g. blogs, emails). Although the two selected treebanks have different annotation schemes, the annotation scheme of the UDT project [205] is based on an evolution of (universal) Stanford dependencies [202]. This allowed us to compare linguistic phenomena correlated with sentence complexity minimizing possible cross-linguistic differences due to not uniform principles of sentence structure representation. In order to reduce the influence of lexicon on the study of sentence complexity, we pruned from the two treebanks those sentences containing low-frequency lemmas with respect to a lemma frequency list that we automatically extracted from a large reference corpus, excluding numerals and proper nouns. For what concerns Italian, we used as a reference corpus PAISÁ [196], which is one of the biggest corpora of authentic contemporary Italian texts. For English, we selected a large corpus of sentences from the Wall Street Journal [223]. For both languages, all the sentences contained in the two treebanks were grouped into 6 bins based on a different sentence length, i.e. 10, 15, 20, 25, 30, 35 tokens (only for Italian with a range of +/-1 tokens each). This was meant to investigate if some linguistic features that are known to correlate with sentence length (e.g. parse tree depth features and dependency links) still play an influence on sentence complexity judgments when sentence length is controlled. Sentences in each subset were then ranked according to the sum of the average frequency of their lemmas. We extracted for each bin the first 200-top ranked sentences, with the exception of Italian for which the last bin contains 123 sentences. As a result of the whole selection process, we obtained 1,200 sentences for English and 1,123 for Italian used for experiments.

4.3.3 Collection of Judgments of Complexity

To collect human complexity judgments, we administered a crowdsourcing task through the platform CrowdFlower[1]. For each language, we recruited 20 native speakers who were asked to read a sentence and rate how difficult it was on a 7-point scale where 1 means "very easy" and 7 "very difficult". Sentences were randomly ordered and presented on distinct pages containing five sentences each. To improve the quality of the collected annotations we chose workers with a "high quality" level assigned by the platform on the basis of their performance in previous tasks and we set a minimum of ten seconds to complete a page. We computed Krippendorff's alpha reliability corresponding to the number of annotators who assigned the same judgment. We obtained reliability of 26% for Italian and 24% for English.

4.4 Studying the Agreement between Human Judgments

Our first research question concerned the investigation of linguistic phenomena characterizing the agreement among annotators in assigning the same judgment of complexity to a sentence. To this end, we split the whole set of rated sentences into ten sets corresponding to the number of annotators giving a judgment of complexity within the same range, hereafter referred to

[1] www.crowdflower.com

degrees of agreement[2]. Figure 4.1 reports the number of sentences for each degree of agreement. For both languages, if we consider a minimum number of 10 agreeing annotators, very few sentences were discarded (~50 for Italian and 70 for English). As the number of agreeing on annotators increases, the number of sentences progressively decreases but we still have a considerable number of sentences (~600) when 14 annotators agree.

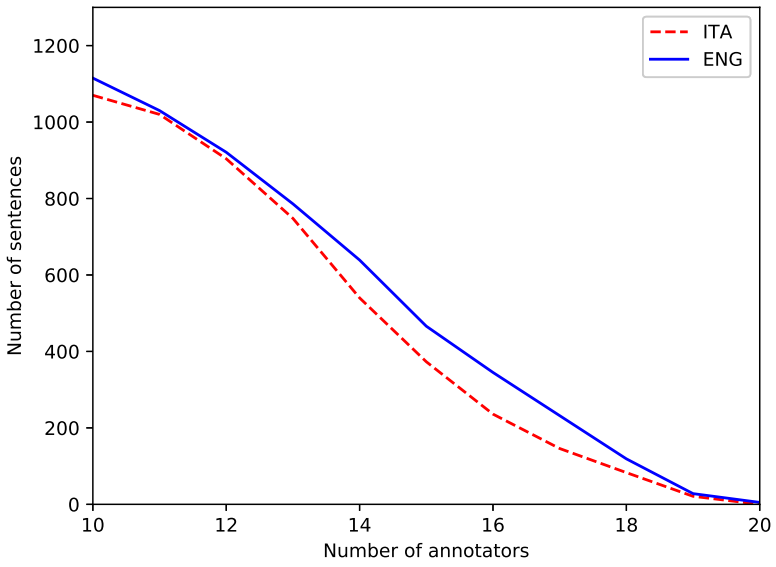


FIGURE 4.1: Number of sentences at different degrees of agreement.

To study the linguistic phenomena characterizing the agreement, we first extracted the features described in Section 4.3.1 from sentences on which annotators agreed (*agreed sentences*) and from the rest of sentences (*not-agreed sentences*); we assessed if the difference is statistically significant using the Wilcoxon Rank-sum test. This was done for each agreement threshold.

We then performed a feature selection process to identify the features that maximize the accuracies of a classifier in predicting *agreed* vs *not-agreed* sentences. To create a ranking of feature relevance, we used the Recursive Feature Elimination (RFE) algorithm implemented in the Scikit-learn library [233], using Linear SVM as an estimator algorithm, and we dropped 1 feature in each iteration. We evaluated the classifier performance using a 3-fold cross-validation method. At the end of this process, we selected the top-ranked features. This procedure was iterated 10 times for each degree of agreement.

In order to evaluate the accuracy of the SVM classifier, we computed a baseline corresponding to the performance of the classifier using a most-likely class classification method, where each sentence is always classified into the most likely class.

Table 4.1 reports the features that vary in a statistically significant way (✓in table) and the ones selected in classification (marked with ⋆) for both languages and degrees of agreement levels. As can be seen, there is an opposite trend between the statistically significant features and those selected by the classifier as the degree of agreement increases. For what concerns

[2]Each range was calculated in terms of standard deviation from the mean judgment values given to each sentence.

the Wilcoxon test, very few features have significantly different values at lower degrees of agreement. That is to say, that very few features are involved in discriminating the *agreed* vs *not-agreed* sentences, especially when the agreement is lower than 14.

For both languages, raw text features (*n_tokens* and *char_tok*) vary significantly at all degrees of agreement. Interestingly, these two features are not considered by the classifier which uses more complex syntactic features, such as features related to subordination (e.g. *subord_depth*) and nominal modification (e.g. *prep_chain_l*). Syntactic features start to vary significantly as the agreement increases, e.g. parse tree depth features such as the depth of the whole parse tree (*max_depth*) and the complement chains (*dep_mark*), and features related to the use of subordination. Comparing the two languages we also found a number of differences. For example, at the lowest agreement (degree 10), features of all types turned out to vary significantly for English, while the Italian *agreed* and *not-agreed* sentences do not vary for any features. At higher agreement, Italian *agreed* sentences are characterized by the variation of two language-specific features: the position of the object with respect to the verb head (*order_obj*) and some verbal morphological features (*verbs_num_pers*, *verbs_tense*), which also contributes to the classification only for Italian.

Table 4.2 reports the accuracy of the SVM classifier for each degree of agreement[3] and the baseline. At lower degrees of agreement (i.e. <14) the classifier achieves lower accuracy compared to the baseline showing that the selected features do not contribute to discriminate *agreed* vs *not-agreed* sentences. Instead, these features start to have a greater impact on the classification of sentences at degrees 14, 15, 16, 17. This means that at these degrees of the agreement the values of the features characterizing the *agreed* sentences are significantly different from those of the *not-agreed* sentences. In addition, even though for these sentences a very high number of features are considered statistically significant by the Wilcoxon test the classifier needs fewer features to assign the correct class (as shown in Table 4.1).

4.5 Correlation of Linguistic Features with Sentence Complexity

The second research question aims to model the human perception of complexity studying the correlation between the set of linguistic features extracted from sentences and the judgments of complexity assigned to each sentence. We first calculated the average complexity judgments for the six bins of sentences of the same length (i.e. 10, 15, 20, 25, 30, 35 tokens). As expected, long sentences were judged as more complex for both languages even though all sentences were always rated as more complex for Italian (see Figure 4.2).

We then calculated the Spearman's rank correlation coefficient between the values of each feature and the average judgments of complexity thus obtaining a ranking of features. The correlation was computed at two distinct degrees of agreement, i.e. 10 and 14. We chose these two thresholds since at 10 the *agreed* sentences correspond to almost all the rated sentences and at 14 the SVM classifier starts to outperform the baseline (see Table 4.2). Besides, at 14 we still have a quite large set of *agreed* sentences allowing a reliable statistical study of the

[3]The accuracy was computed as the average classification score of the 10 best results of the feature selection process.

Feature	Agreement															
	10		11		12		13		14		15		16		17	
	IT	EN	IT	EN	IT	EN	IT	EN	IT	EN	IT	EN	IT	EN	IT	EN
char_tok	★	★	★	★	★	-	-	✓	✓	✓★	✓★	✓★	✓★	✓★	✓★	✓★
cpos_ADJ	★	★	★	★	★	-	-	-	✓★	-	✓★	✓★	✓★	✓★	✓★	✓
cpos_ADP	★	★	★	★	★	-	-	★	-	-	-	-	✓	✓	✓	✓
cpos_ADV	★	-	★	-	★	-	-	-	-	-	★	-	★	-	★	-
cpos_AUX	★	-	★	-	★	-	✓	-	-	-	-	-	✓★	-	✓	-
cpos_CONJ	★	★	★	★	★	-	-	★	-	✓	✓	✓★	✓★	✓	✓	✓
cpos_PRON	★	-	★	-	★	-	-	-	✓	-	✓★	-	✓	-	✓	-
cpos_DET	-	★	-	★	-	-	-	✓★	-	✓★	-	✓★	-	✓★	-	✓★
cpos_NUM	-	★	-	✓★	-	✓	-	✓★	-	✓★	-	✓★	-	✓★	-	✓★
cpos_PROPN	★	-	★	-	★	-	-	-	✓	-	★	-	✓★	-		-
cpos_PUNCT	★	-	★	-	★	-	✓	-	-	-	✓★	-	✓	-	✓★	-
cpos_SCONJ	★	-	★	-	★	-	-	-	-	-	✓★	-	✓	-	✓	-
cpos_VERB	-	★	-	★	-	✓	-	✓★	-	✓★	-	✓★	-	✓★	-	✓
dep_acl	-	-	★	-	★	-	✓	-	✓	-	✓	-	✓	-	✓	-
dep_acl:relcl	-	-	★	-	★	-	-	-	★	-	✓	-	✓★	-	✓	-
dep_adpobj	-	★	-	★	-	-	-	★	-	-	-	-	-	-	-	✓
dep_advcl	★	-	★	-	★	-	-	-	✓	-	✓	-	✓★	-	✓	-
dep_amod	★	✓★	★	★	★	✓	-	✓★	✓	✓	✓	✓★	✓★	✓★	✓	✓★
dep_appos	-	★	-	★	-	-	-	-	-	★	-	-	-	-	-	-
dep_attr	-	★	-	★	-	-	-	-	-	-	-	✓★	-	✓★	-	✓
dep_aux	-	-	★	-	★	-	✓	-	✓	-	-	-	✓★	-	✓	-
dep_case	★	-	★	-	★	-	-	-	★	-	-	-	✓	-	✓	-
dep_cc	★	★	★	★	★	-	-	-	-	✓★	✓	✓★	✓★	✓★	✓	✓
dep_ccomp	-	★	-	★	-	-	-	-	-	✓	-	✓	-	✓	-	✓
dep_compmod	-	★	-	★	-	-	-	-	-	✓★	-	★	-	✓★	-	✓★
dep_conj	★	★	★	★	★	-	-	✓★	-	✓★	✓★	✓★	✓★	✓★	✓	✓★
dep_det	-	★	-	★	-	-	-	★	-	✓★	-	✓★	-	✓★	-	✓★
dep_dobj	★	-	★	-	★	-	-	-	-	-	✓	-	✓★	-	✓	-
dep_mark	★	★	★	★	★	-	✓	★	✓	★	✓★	★	✓★	✓	✓	✓
dep_nmod	★	★	★	★	✓★	-	✓	-	✓	-	-	✓★	✓★	✓	✓	✓
dep_nsubj	-	✓★	-	✓★	-	✓	-	✓	-	✓★	-	✓★	-	✓★	-	✓★
dep_num	-	★	-	★	-	✓	-	✓	-	✓★	-	✓★	-	✓★	-	✓★
dep_partmod	-	★	-	★	-	-	-	-	-	-	-	✓	-	✓	-	✓
dep_poss	-	★	-	★	-	-	-	-	-	✓	-	✓	-	✓	-	✓
dep_punct	★	-	★	-	★	-	✓	-	-	-	✓★	-	✓	-	✓	-
dep_rcmod	-	★	-	★	-	-	-	★	-	-	-	✓★	-	✓★	-	✓
dep_xcomp	★	-	★	-	★	-	-	-	-	-	-	-	✓	-	✓	-
lex_density	-	★	-	★	-	-	-	✓★	-	✓★	-	✓	-	✓★	-	✓★
links_len	-	✓★	★	★	✓★	✓	✓	✓	✓★	✓	✓	✓★	✓	✓	✓	✓
max_depth	-	★	★	★	✓★	-	✓	✓	✓	✓	✓	✓	✓	✓★	✓	✓
max_links_l	-	✓★	★	✓★	✓★	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
n_prep_chains	★	✓★	✓★	★	✓★	✓	✓	✓	✓	✓	✓	✓	✓	✓★	✓	✓
n_principal_clauses	-	★	★	★	★	-	✓	✓	✓	✓★	✓	✓	✓	✓	✓	✓
n_subord_chain	★	★	★	★	★	✓	✓	-	✓★	✓★	✓★	✓	✓★	✓	✓	✓
n_subord_clauses	★	-	★	-	★	-	✓	-	✓★	-	✓★	-	✓★	-	✓★	-
n_tokens	-	✓★	✓★	✓★	✓★	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
order_obj	-	-	★	-	★	-	-	-	-	-	✓	-	✓	-	✓	-
order_subj	-	-	★	-	★	-	-	-	★	-	-	-	✓	-	✓	-
order_subord	★	★	★	★	★	-	✓	✓	✓	✓★	✓	✓	✓	✓	✓	✓
prep_chain_l	-	★	★	★	★	-	✓	-	✓	✓	✓★	✓	✓★	✓	✓	✓
prep_depth	-	✓★	★	★	✓	✓	✓	✓	✓	✓★	✓	✓★	✓	✓★	✓	✓★
subord_depth	★	★	★	★	★	-	✓★	★	✓★	✓★	✓★	✓★	✓★	✓★	✓★	✓★
token_clause	-	★	★	★	★	-	-	-	-	-	-	-	✓	✓	✓	✓
ttr_form	-	✓★	★	★	✓★	✓	✓	✓	✓	✓★	✓★	✓★	✓★	✓★	✓	✓★
ttr_lemma	★	✓★	★	✓★	★	✓	✓	✓★	✓	✓★	✓★	✓★	✓★	✓★	✓★	✓★
verb_arity	★	★	★	★	✓★	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
verb_head_arity	★	★	★	★	★	★	★	★	✓★	★	✓★	✓★	✓★	✓★	✓★	✓★
verb_head	★	★	★	★	✓★	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
verbs_num_pers	★	-	★	-	★	-	✓★	-	✓★	-	✓★	-	✓★	-	✓★	-
verbs_tense	★	-	★	-	✓★	-	✓	-	✓★	-	✓★	-	✓★	-	✓★	-

TABLE 4.1: Linguistic features that vary statistically (✓) and the ones selected by the SVM classifier in at least 50% of the 10 runs (★) for Italian and English at different degrees of agreement.

features (see Figure 4.1). Only at threshold 10 we also calculated the ranking of the features with respect to the six bins of sentences of the same length (L10, L15, L20, L25, L30, L35). Figure 4.3 reports the ranking of features with p <0.05. Positive numbers mean that the higher the feature value the more complex the sentence was judged (i.e. the feature ranked +1 is the top-ranked one since it is the most positively correlated). Instead, negative numbers mean that the lower the feature value, the more complex the sentence was judged (i.e. the feature ranked -1 is the highest negatively correlated). In both languages, the correlation between the top 20 ranked features and the complexity judgment is extremely high, ranging from 0.30 to 0.85 when we consider sentences at agreement 14. At the two agreement thresholds, for all lengths (columns *T10*, *T14*), they concern not only sentence length but also deep syntactic features, in terms of e.g. the depth of the whole parse tree (*max_depth*), the length of dependency links

	Baseline Accuracy (%) – SVM Classifier Accuracy (%)							
	10	11	12	13	14	15	16	17
Italian	95.4-95.4	91-90.8	80.6-80.5	66.7-66	51.9-**59.1**	66.8-**68.8**	79-**80.7**	87-**87.1**
English	94-94	86.8-86.8	83.6-77.4	66.3-66.1	53.9-**60**	60.7-**71.8**	70.9-**79.3**	80.4-**84.6**

TABLE 4.2: Baseline and SVM classifier accuracy at different degrees of human agreement.

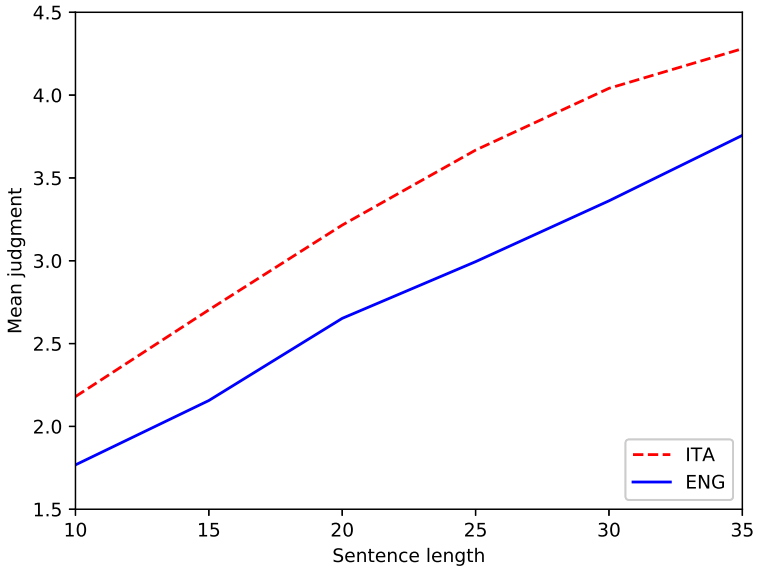
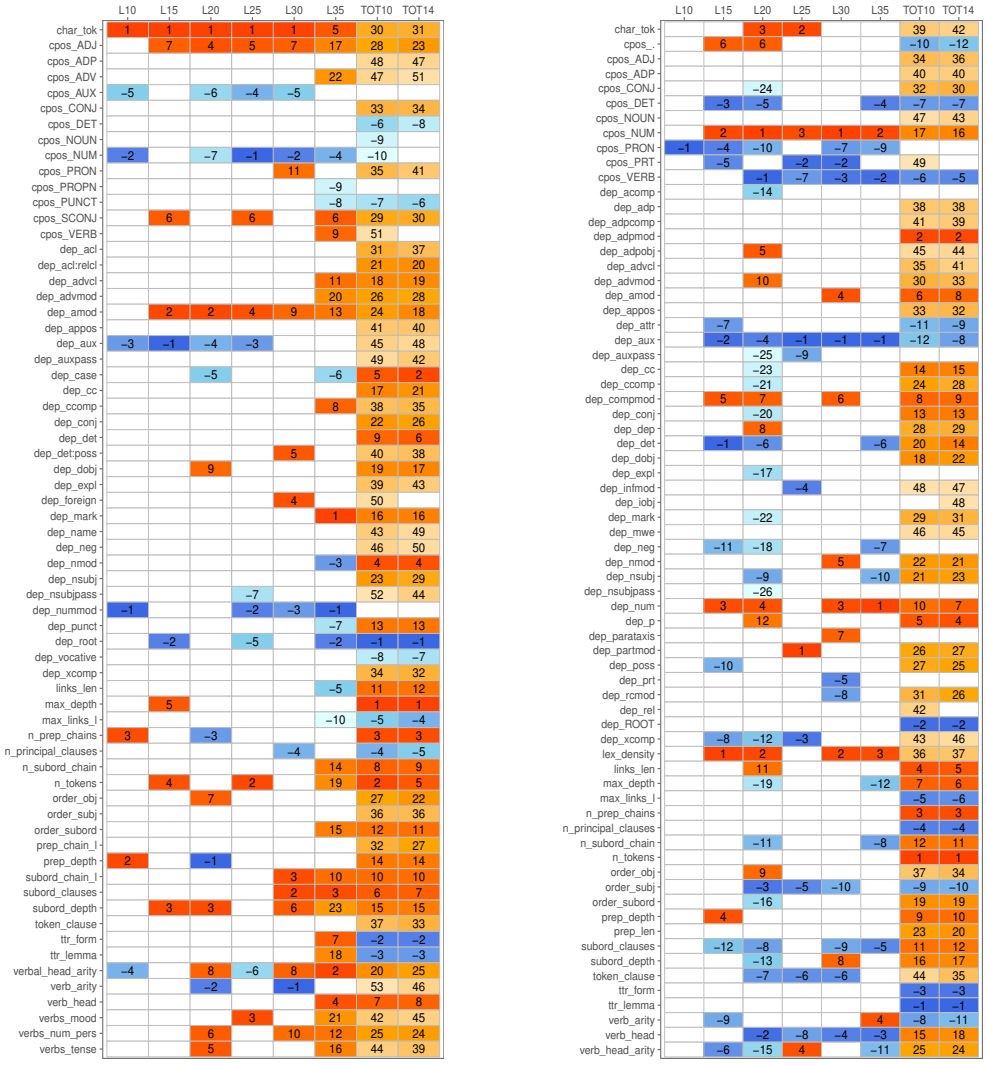


FIGURE 4.2: Mean complexity judgment at different sentence length.

(*links_len*), and features related to subordination (e.g. *n_subord_clauses*). Specifically, the 1st-ranked feature in Italian (parse tree depth) and the one in English (sentence length) have a correlation of 0.64 and 0.84 respectively. Nominal modification (*n_prep_chains*) is also highly correlated (Italian r_s=0.59, English 0.54) and similarly ranked in the two languages at 3rd position. The distribution of *verbs_num_pers* makes the sentence harder only for Italian; this is possibly related to the higher complexity of verbal morphology since the 3rd person verbs in impersonal structures might increase the ambiguity of the sentence with respect to the referent. Only in English, sentence complexity is affected by the distribution of cardinal numbers (*cpos_NUM*) and the dependency type "numeric modifier" (*dep_num*), in line with the difficulty of numerical information shown in readability studies [21]. Conversely, the verbal arity and the relative ordering of subjects with respect to the verb have a lower position in the negative ranking, suggesting that these features make a sentence easier: this might be due to a more fixed predicate-argument structure and word order in this language.

If we focus on sentences of the same length, features considered as a proxy of lexical complexity are in the top positions in both languages. It is the case of the average word length (*char_tok*) and the lexical density (*lex_density*) only for English. Interestingly, while for English the majority of features are similarly ranked in all bins of sentences of the same length, for Italian we observe differences between the rankings of features extracted from sentences $\leq$ and $\geq$20 token long. Namely, when the average sentence length is $\geq$20 tokens, features related to subordination make the sentence more complex.



(a) Italian. (b) English.

FIGURE 4.3: Features correlating with human judgments at different sentence lengths and with respect to the sentences at agreement 10 (*TOT 10*) and 14 (*TOT 14*).

4.5.1 Predicting Human Complexity Judgments

To assess the contribution of the linguistic features to predict the judgment of sentence complexity we trained a linear SVM regression model with default parameters. We performed 3-fold cross-validation over each subset of *agreed* sentences at agreement 10 and 14. We measured two performance metrics: the *mean absolute error* to evaluate the accuracy of the model to predict the same complexity judgment assigned by humans; the *Spearman correlation* to evaluate the correlation between the ranking of features produced by the regression model with the ranking produced by the human judgments. Table 4.3 reports the results and the average score of the two metrics. As it can be seen, the model is very accurate and achieves a very high correlation (>0.56 with p <0.001) with an average error difference (*avg mean abs err*) below 1. In particular, the model obtained higher performance in predicting the ranking of features extracted from sentences at agreement 14. This might be

due to the fact these sentences are characterized by a more uniform distribution of linguistic phenomena and that these phenomena contribute to predict the same judgment of complexity. This is in line with the results obtained by the SVM classifier in predicting agreement (Table 4.2). This is particularly the case of English and it possibly suggests that the set of sentences similarly judged by humans are characterized by a lower variability of the values of the features.

	IT-10	**IT-14**	**EN-10**	**EN-14**
mean abs err 1	0.77	0.78	0.71	0.68
Spearman 1	0.57	0.64	0.68	0.64
mean abs err 2	0.79	0.80	0.70	0.70
Spearman 2	0.55	0.63	0.67	0.73
mean abs err 3	0.85	0.75	0.77	0.60
Spearman 3	0.55	0.64	0.61	0.71
avg mean abs err	**0.80**	**0.78**	**0.72**	**0.66**
avg Spearman	**0.56**	**0.63**	**0.65**	**0.69**

TABLE 4.3: Performance of the linear SVM regression model and the avg score at different agreements.

4.6 Discussion and Conclusion

We introduced a method to model the human perception of sentence complexity relying on a new corpus of Italian and English sentences rated with human complexity judgments. We tested the contribution of a wide set of linguistic features automatically extracted from these sentences in two experimental scenarios. The first one highlighted that we can reliably predict the degree of agreement between human annotators, independently from the assigned judgment of complexity: given the high subjectivity of the task, this is a quite notable result that to our knowledge has never been reported. We observed in particular that deep syntactic features related to e.g. the use of subordination and nominal modification play a main role in the automatic prediction of human agreement. This is true for the two languages even though we found that some features resulted to be more relevant in the classification of *agreed* Italian sentences, e.g. the relative ordering of the object. Interestingly, we also noticed that the classifier needs a few features to predict agreed sentences when more than half of the annotators shares the same judgment.

In the second experiment, we studied the correlation between linguistic features and complexity judgments. The resulting ranking highlighted the key role played by syntactic phenomena: features related to sentence structure are among the top-ranked features characterizing sentences that were rated highly complex by a given number of agreeing on annotators. When sentence length was controlled, the relevance of the considered features changes in particular for Italian: e.g. features concerning the use of subordination make the sentence more complex when sentence length is $\geq$20 tokens. As shown by the results of the regression model, the set of studied features contribute significantly to automatically predict the human judgment of sentence complexity.

In addition, the presented corpus can be useful for different applications. From a psycholinguistic perspective, it can be used for comparison with data collected through controlled experimental scenarios assessing sentence complexity in terms of cognitive measures (offline and online), which are also more constrained and costly to acquire in large-dimensions. The corpus also allows studying whether features of linguistic complexity are implied in modelling other properties of texts, such as the level of engagement or subjectivity. From an NLP perspective, the corpus can be exploited to train systems able to predict people's perception of complexity. For example, it can support a range of related tasks, such as the development of linguistically-informed algorithms for the automatic assessment of text difficulty, as well as in Natural Language Generation tasks, going from text simplification to the automatic generation/evaluation of highly-engaging texts.

Chapter 5

Modelling Style and Affects in Natural Language

5.1 Introduction

In this chapter, we focus on NLU techniques to model stylistic and affects aspects of language. A special focus is dedicated to Multi-Task Learning (MTL). Some of the studies reported in this chapter were carried out in the context of the Evaluation of NLP and Speech Tools for Italian (EVALITA) 2018 [44] and 2020 [20] workshops. The objective of the tasks is sentiment analysis, irony detection, hate speech recognition and author profiling. The proposed systems outperformed the state of the art for each task in which we participated; in 2018, the proposed system also obtained the Best System Award[1].

The chapter is structured in three main sections, the first one is focused on preliminary experiments carried out on the SENTIPOLC shared task dataset [11]. The second and the third reports the experiments carried out in the scope of EVALITA 2018 and EVALITA 2020 respectively.

5.2 Multi-Task Learning in Deep Neural Network for Sentiment Polarity and Irony classification

One of the aspects we analyzed is Sentiment Analysis and related tasks. Several works have been published during the last years on these topics and, with the rising of deep learning, the performances of the systems have considerably increased. Despite these performances improvements, machine learning-based systems still struggle to perform well in edge cases such as when literal polarity is inverted by irony, especially when these cases are under-represented in the training data. Such cases were annotated for the SENTIPOLC 2016 shared task [11]: consider the tweet from the dataset *"Ho molta fiducia nel nuovo Governo Monti. Più o meno la stessa che ripongo in mia madre che tenta di inviare un'email"* ("I have a lot of faith in the new Monti government. More or less the same thing that I have in my mother who tries to send an email"): this tweet has literal positive polarity, but irony changes the final polarity annotation.

[1] http://www.evalita.it/2018/best-system

Previous works on neural networks already shown issues on learning such difficult cases: [201] pointed out a set of 10 criticisms of deep neural networks like the inability to deal with a hierarchical structure, the limited capacity for transfer learning, the impossibility to integrate prior knowledge or lack of systematic compositional skills. Despite these issues, previous works [281] have shown that multi-task learning (MTL) is an appealing idea compared to single-task learning (STL) since it allows to incorporation of previous knowledge about tasks hierarchy into neural networks architectures. [262] have shown that MTL is useful to combine even loosely related tasks, letting the networks automatically learn the tasks hierarchy.

To study the effectiveness of MTL on Sentiment Analysis tasks, in [76] we introduced a mixed MTL/STL approach (named MIX) based on deep bi-directional recurrent neural networks [270] applied to polarity and irony detection on Italian tweets. We modelled our networks to solve three binary tasks: positive, negative and ironic tweet identification. We tested the performances of our system on the most recent datasets available for Italian (at the time in which the study has been done). We show that our system outperforms the state of the art for Italian for what concerns polarity and irony classification. Furthermore, we show that the proposed mixed approach outperforms both our STL and MTL approaches.

To our knowledge, this is the first work that shows the effectiveness of MTL combining irony and polarity detection. A previous work on this topic [84] has been presented at EVALITA 2016, but the authors proposed an approach that is more similar to a multi-label classification method based on a single classifier for all the labels, rather than an MTL in which different loss functions are used for the different tasks.

We present an in-depth analysis of the results obtained by our method showing how the proposed multi-task learning approach is able to compose the information coming from the different tasks.

Our contributions: (i) to our knowledge this is the first work that presents an MTL system for polarity and irony detection; (ii) we introduce a novel mixed MTL and STL approach; (iii) we present an error analysis that suggests that the proposed multi-task learning approach is able to combine the information extracted from sentiment polarity and irony classification training set and improves the performance on both the tasks. This is particularly true on edge cases in which knowledge about the two tasks are needed to classify a tweet.

Successive studies such as [314] further demonstrated that Irony Detection can benefit from sentiment-based transfer learning.

5.2.1 Dataset

For the Italian polarity and irony classification tasks we relied on the dataset provided for the SENTIPOLC event which made part of EVALITA 2016, the periodic evaluation campaign NLP and speech tools for the Italian language. The SENTIPOLC dataset contains a training set made of 7,410 tweets and a test set of 2,000 tweets. Each tweet was labelled with a set of 6 binary labels that define if a tweet is subjective (*subj*), positive (*pos*), negative (*neg*), ironic (*iro*), literally positive (*lpos*) and literally negative (*lneg*). We performed our experiments only on positive, negative and ironic classes, but we still used the other labels to perform a

Label combination						train	test
subj	pos	neg	iro	lpos	lneg		
0	0	0	0	0	0	2,312	695
1	0	0	0	0	0	504	219
1	1	0	0	1	0	1,488	295
1	0	1	0	0	1	1,798	520
1	1	1	0	1	1	440	36
1	0	1	1	0	0	210	73
1	1	0	1	1	0	62	8
1	0	1	1	1	0	239	66
1	1	0	1	0	0	29	3
1	0	1	1	0	1	225	53
1	1	0	1	0	1	22	4
1	0	1	1	1	1	71	22
1	1	0	1	1	1	10	6
total						7,410	2,000

TABLE 5.1: Distribution of label combinations in the SENTIPOLC 2016 data set

comparative analysis between the performances of the system trained in the single-task and in the multi-task models.

Table 5.1 reports the distributions of labels in the data set.

5.2.2 Architecture and Training

Figure 5.1 reports the architectures of the MTL and STL neural networks that we designed. Both the architectures are based on bidirectional long short-term memory networks (Bi-LSTM) [135, 122]. The STL architecture is composed of two stacked Bi-LSTM layers and a dense layer for each task. The MTL architecture is composed of a *shared Bi-LSTM*, three task-specific Bi-LSTMs, and three task-specific dense layers specialized in recognizing respectively positive, negative and ironic inputs. We introduce in this work a new method

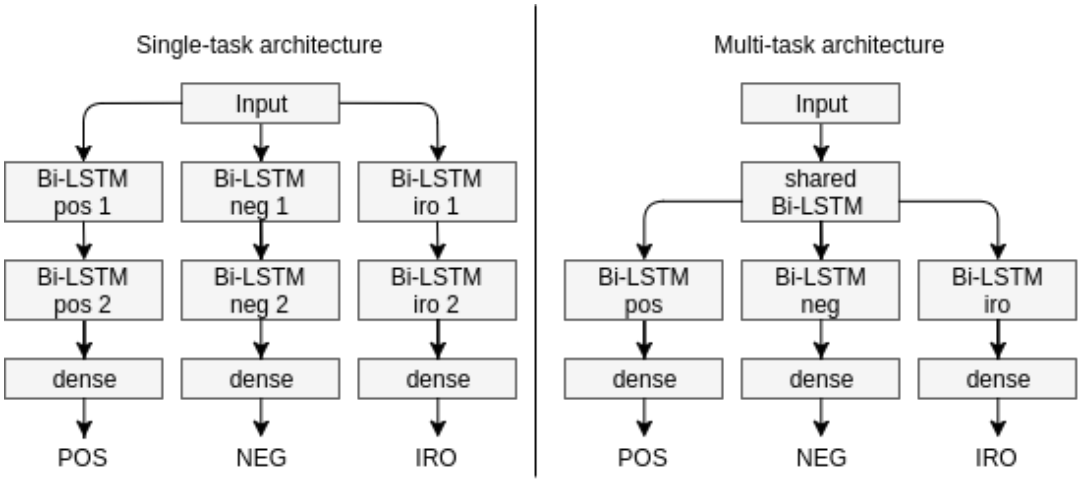


FIGURE 5.1: STL and MTL neural networks architectures.

(named MIX) to combine these two architectures using a two-stage training approach in which a layer is shared in just one stage of the training phase.

Features: We built two sets of word embeddings with 128 dimensions using word2vec [211]. The first set of word embeddings was generated starting from the itWac Corpus [15], while the second was built exploiting approximatively 25 millions of Italian tweets. Both the corpora were postagged using the postagger by [55] and the word embeddings were computed using the combination of the word and its part of speech. The generated itWac and Twitter embeddings provided coverage of 91.5% and 96.6% on the SENTIPOLC dataset. In addition, for each word its sentiment polarity is used as a feature exploiting the sentiment polarity lexicon by [198].

Each token of a tweet is represented by a vector resulting from the concatenation of the described features.

Training: To train the STL networks, we performed three different training steps, one for each task. To train the MTL architecture, we run a shared training by iteratively optimizing at each step a loss function for each task. For the MTL the global loss function is given by the sum of the three individual loss functions. In STL and MTL architectures, we stopped the training after 50 epochs without improvements of the loss function on the validation set, choosing the parameters with the best performances.

To mix the MTL and STL approaches we used a two-stage training. In the first stage, we trained the MTL network as described above. In the second stage, we initialized the weights of the three first Bi-LSTM layers of the STL architecture using the weights of the MTL network's *shared Bi-LSTM* and the second level Bi-LSTM using the weights learned in the first stage. We then run specific training for each task. We used the same stopping criteria as for STL and MTL training.

Since in the dataset all the tweets are labelled with their polarity and irony labels and the number of ironic tweets is extremely unbalanced w.r.t. the non-ironic ones, we oversampled the ironic examples by replicating them in the dataset. The oversampling technique has been showing to improve classification performance on unbalanced datasets [48].

5.2.3 Results

Table 5.2 reports the performances on the test set achieved by our baselines and multi-task learning models. The scores are calculated accordingly to the official metrics adopted by the task organizers. Since random initialization leads to different performances in different runs, we repeated the experiments 10 times and the tables report the average scores. In addition, the tables report the performances obtained by the best systems that participated in SENTIPOLC 2016. To study the impact of multi-task learning across irony and polarity tasks, we also tested a MIX model trained only on positive and negative labels (PMIX) without using irony information.

As we can see in Table 5.2, in the polarity detection tasks the MTL, PMIX, and MIX models all outperform the best SENTIPOLC system that used a single task approach [6] (UniPI.2.c row), while only the MIX model performed better than the [84] system (SwissChese.c row), that used a multi-label classifier for the subjectivity, polarity and irony identification tasks.

System	POS	NEG	Polarity	IRO
STL	.641	.665	.653	.608
PMIX	.670	.699	.684	-
MTL	.674	.700	.674	.586
MIX	.660	**.736**	**.698**	**.622**
SwissCheese.c	.653	.713	.683	.536
UniPI.2.c	**.685**	.643	.664	-
tweet2check16.c	-	-	-	.541

TABLE 5.2: F1-Scores obtained on the SENTIPOLC 2016 dataset by the different systems. *Polarity* is the official metric for the polarity detection task and it is a combination of POS and NEG accuracies.

System	POS		NEG		Polarity	
	Iro	l_Pol	Iro	l_Pol	Iro	l_Pol
STL	.115	.105	0.11	.090	.080	.085
PMIX	.143	.044	.093	.075	.093	.049
MTL	.104	.069	.086	.075	.086	.061
MIX	**.539**	**.567**	**.553**	**.492**	**.553**	**.500**

TABLE 5.3: Polarity F1-Scores for ironic tweets (Iro) and for tweets in which irony modifies literal polarity (l_Pol) in the Italian test set.

Label combination						Freq	IRO Accuracy		
subj	pos	neg	lpos	lneg	iro		MIX	MTL	STL
1	1	0	0	0	1	8	**37.50**	0.00	0.00
1	1	0	0	0	1	3	**33.33**	0.00	0.00
1	1	0	0	1	1	4	**50.00**	0.00	0.00
1	1	0	1	1	1	6	16.67	0.00	**33.33**
1	0	1	1	1	1	22	**27.27**	4.55	13.64
1	0	1	1	0	1	66	**21.21**	6.06	4.55
1	0	1	0	0	1	73	**32.88**	4.11	8.22
1	0	1	0	1	1	53	**26.42**	5.66	5.66
1	1	0	1	0	0	295	**94.58**	93.22	91.19
1	1	1	1	1	0	36	80.56	**97.22**	**97.22**
1	0	1	0	1	0	520	89.81	**97.31**	94.04
0	0	0	0	0	0	695	**98.27**	95.83	93.38
1	0	0	0	0	0	219	92.24	**95.43**	93.61

TABLE 5.4: Irony accuracy of our models for the different combinations of labels in the SENTIPOLC 2016 test set.

For what concerns Irony detection, we observe that all our networks outperform the best SENTIPOLC system, probably thanks to the usage of oversampling (the F-score of our STL model without oversampling is only 0.473). More importantly, we observe that the MIX model significantly outperforms the STL baseline, while the standard MTL does not. These results show that the MIX model brings improvement in both polarity and irony detection tasks.

To study the impact of multi-task learning in Polarity and Irony detection, we conducted an in-depth error analysis to investigate the performance of our models on edge cases. We studied the behaviour of the models for a selected subset of the test set. Table 5.3 reports the polarity detection accuracies of our models on Italian ironic tweets (columns *Iro* in the table) and on tweets for which irony changes the literal polarity (*l_Pol*). We can clearly observe how the MIX model brings great improvements for polarity detection in *l_Pol* tweets while the standard MTL does not. The improvements are clear for both positive and negative tweets. This result suggests that the MIX model is able to compose information coming from different examples of different tasks and to obtain better results on edge cases. This is also shown by the results obtained in the polarity detection task on ironic tweets (*Iro*).

Table 5.4 reports the accuracy of our systems in the irony detection task for all the different label combinations in the test set. We can observe that the STL and the MTL models show the same behaviour while the MIX model significantly outperforms the other two in mostly all kinds of ironic instances (rows 1-8) and not ironic positive instances (row 9). Vice versa, MTL and STL outperform MIX in the negative, not ironic comments (rows 10-11). Given that the MIX approach brings impressive improvements for edge-cases (especially rare ones), it is likely that it overestimates the correlation between irony and negativity.

5.2.4 Conclusion

We conducted a study on the effectiveness of multi-task learning approaches in sentiment polarity and irony classification. We presented a mixed single- and multi-task learning approach, that is able to improve the performance both in polarity and irony detection with respect to single-task and standard multi-task learning approaches. In particular, our approach led to substantial improvements on edge cases in which knowledge about the two tasks are needed to classify a tweet. This is particularly true when these cases are under-represented in the training data. An example is a case when the literal polarity of a tweet is inverted by irony. However, the performances of all the systems are far to be enough reliable to be exploited for NLG systems evaluation.

5.3 Multi-Task Learning at EVALITA 2018

The EVALITA 2018 edition has been one of the most successful editions in terms of the number of shared tasks proposed. In particular, a large part of the tasks proposed by the organizers can be tackled as binary document classification tasks. This gave us the possibility to test a new system specifically designed for this EVALITA edition. In this context in [58] we introduced a system which relies on Bi-LSTM [135, 122] and SVM which are widely used learning algorithms in the document classification task. The learning algorithm can be selected in a configuration file. In this work, we used the Keras[2] library and the liblinear[3] library to generate the Bi-LSTM and SVM statistical models respectively. Since our

[2]https://github.com/keras-team/keras
[3]https://www.csie.ntu.edu.tw/ cjlin/liblinear/

approach relies on morphosyntactically tagged text, training and test data were automatically morphosyntactically tagged by the PoS tagger described in [55].

Some specific adaptions were made due to the characteristics of each shared task. In the Aspect-based Sentiment Analysis (ABSITA) 2018 shared task [18] participants were asked, given a training set of Booking hotel reviews, to detect the mentioned aspect categories in a review among a set of 8 fixed categories (ACD task) and to assign the polarity (neutral, positive, neutral, positive-negative) for each detected aspect (ACP task). Since each Booking review in the training set is labelled with 24 binary labels (8 indicating the presence of an aspect, 8 indicating positivity and 8 indicating negativity w.r.t. an aspect), we addressed the ABISTA 2018 shared task as 24 binary classification problems.

The Gender X-Genre (GxG) 2018 shared task [81] consisted of the automatic identification of the gender of the author of a text (Female or Male). Five different training sets and test sets were provided by the organizers for five different genres: Children essays (CH), Diary (DI), Journalism (JO), Twitter posts (TW) and YouTube comments (YT). For each test set the participants are requested to submit a system trained using an in-domain training dataset and a system trained using cross-domain data only.

The IronITA task [54] consisted of two tasks. In the first task, participants had to automatically label a message as ironic or not. The second task had a more fine-grain: given a message, participants had to classify whether the message is sarcastic, ironic but not sarcastic or not ironic.

Finally in the HaSpeeDe 2018 shared task [31] consisted of automatically annotating messages from Twitter and Facebook with a boolean value indicating the presence (or not) of hate speech. In particular, three tasks were proposed: `HaSpeeDe-FB` where only the Facebook dataset could be used to classify Facebook comments, `HaSpeeDe-TW` where just Twitter data could be used to classify tweets and `Cross-HaspeeDe` where only the Facebook dataset could be used to classify the Twitter test set and vice versa (`Cross-HaspeeDe_FB`, `Cross-HaspeeDe_TW`).

5.3.1 Lexical Resources

Automatically Generated Sentiment Polarity Lexicons for Social Media: For the purpose of modelling the word usage in generic, positive and negative contexts of social media texts, we developed three lexicons which we named $TW_{GEN}, TW_{NEG}, TW_{POS}$. Each lexicon reports the relative frequency of a word in three different corpora. The main idea behind building these lexicons is that positive and negative words should present a higher relative frequency in TW_{POS} and TW_{NEG} respectively. The three corpora were generated by first downloading approximately 50,000,000 tweets and then applying some filtering rules to the downloaded tweets to build the positive and negative corpora (no filtering rules were applied to build the generic corpus). In order to build a corpus of positive tweets, we constrained the downloaded tweets to contain at least one positive emoji among heart and kisses. Since emojis are rarely used in negative tweets, to build the negative tweets corpus we created a list of commonly used words in negative language and constrained these tweets to contain at least one of these words.

Automatically translated Sentiment Polarity Lexicons: The Multi–Perspective Question Answering (hereafter referred to as *MPQA*) Subjectivity Lexicon [306]. This lexicon consists of approximately 8,200 English words with their associated polarity. To use this resource for the Italian language, we translated all the entries through the Yandex translation service[4]. We used the Yandex service instead of others such as Google Translate because they provide higher rate limits, allowing us to translate the full lexicon for free.

Word Embedding Lexicons: We generated four word embedding lexicons using the word2vec[5] toolkit [211]. As recommended in [211], we used the CBOW model that learns to predict the word in the middle of a symmetric window based on the sum of the vector representations of the words in the window. For our experiments, we considered a context window of 5 words. The Word Embedding Lexicons starting from the following corpora which were tokenized and postagged by the PoS tagger for Twitter described in [55]:

- The first lexicon was built using the itWaC corpus[6]. The itWaC corpus is a 2 billion word corpus constructed from the Web limiting the crawl to the .it domain and using medium-frequency words from the Repubblica corpus and basic Italian vocabulary lists as seeds.
- The second lexicon was built using the set of the 50,000,000 tweets we downloaded to build the sentiment polarity lexicons.
- The third and the fourth lexicon were built using a corpus consisting of 538,835 Booking reviews scraped from the web. Since each review in the Booking site is split in a positive section (indicated by a plus mark) and negative section (indicated by a minus mark), we split these reviews obtaining in 338,494 positive reviews and 200,341 negative reviews. Starting from the positive and the negative reviews, we finally obtained two different word embedding lexicons.

Each entry of the lexicons maps a pair (word, POS) to the associated word embedding, allowing to mitigation polysemy problems which can lead to poorer results in classification. In addition, both the corpora were preprocessed in order to 1) map each URL to the word "URL" 2) distinguish between all uppercased words and non-uppercased words (eg.: "mai" vs "MAI"), since all uppercased words are usually used in negative contexts. Since each task has its own characteristics in terms of information that needs to be captured from the classifiers, we decided to use a subset of the word embeddings in each task. Table 5.5 sums up the word embeddings used in each shared task.

5.3.2 The Classifier

The classifier we built for our participation in the tasks was designed with the aim of testing different learning algorithms and learning strategies. More specifically our classifier implements two workflows that allow testing SVM and recurrent neural networks as learning

[4]http://api.yandex.com/translate/
[5]http://code.google.com/p/word2vec/
[6]http://wacky.sslmit.unibo.it/doku.php?id=corpora

Task	Booking	ITWAC	Twitter
ABSITA	✓	✓	✗
GxG	✗	✓	✓
HaSpeeDe	✗	✓	✓
IronITA	✗	✓	✓

TABLE 5.5: Word embedding lexicons used by our system in each shared task (✓used; ✗not used).

algorithms. In addition, when recurrent neural networks are chosen as learning algorithms, our classifier allows performing neural network multi-task learning (MTL) using an external dataset in order to share knowledge between related tasks. We decided to test the MTL strategy since, as demonstrated in [76], it can improve the performance of the classifier on emotion recognition tasks. The benefits of this approach were investigated also by [281], which showed that MTL is appealing since it allows to incorporate previous knowledge about tasks hierarchy into neural networks architectures. Furthermore, [262] showed that MTL is useful to combine even loosely related tasks, letting the networks automatically learn the tasks hierarchy.

Both the workflows we implemented to share a common pattern used in machine learning consisting of a document feature extraction and a training phase based. However, since the SVM and the Bi-LSTM take input 2-dimensional and 3-dimensional tensors respectively, two different feature extraction procedure are used. In addition, when the Bi-LSTM workflow is selected the classifier can take as input an extra file that will be used to exploit the MTL learning approach. Furthermore, when the Bi-LSTM workflow is selected, the classifier performs a 5-fold training approach. More precisely we build 5 different models using different training and validation sets. These models are then exploited in the classification phase: the assigned labels are the ones that obtain the majority among all the models. The 5-fold approach strategy was chosen in order to generate a global model which should less be prone to overfitting or underfitting w.r.t. a single learned model.

The SVM classifier: The SVM classifier exploits a wide set of features ranging across different levels of linguistic description. With the exception of the *word embedding combination*, these features were already tested in a previous participation at the EVALITA 2016 SENTIPOLC edition [56]. The features are organised into three main categories: *raw and lexical text features*, *morpho-syntactic features* and *lexicon features*. Due to size constraints, we report only the feature names.

Raw and Lexical Text Features number of tokens, character *n*-grams, word *n*-grams, lemma *n*-grams, repetition of *n*-grams chars, number of mentions, number of hashtags, punctuation.

Morpho-syntactic Features coarse grained Part-Of-Speech *n*-grams, Fine grained Part-Of-Speech *n*-grams, Coarse grained Part-Of-Speech distribution

Lexicon features Emoticons Presence, Lemma sentiment polarity *n*-grams, Polarity modifier, PMI score, sentiment polarity distribution, Most frequent sentiment polarity, Sentiment polarity in text sections, Word embeddings combination.

The Deep Neural Network classifier: We tested two different models based on Bi-LSTM: one that learns to classify the labels without sharing information from all the labels in the training phase (Single task learning - STL), and the other one which learns to classify the labels exploiting the related information through a shared Bi-LSTM (Multi-task learning - MTL). We employed Bi-LSTM architectures since these architectures allow us to capture long-range dependencies from both directions of a document by constructing bidirectional links in the network [270]. We applied a dropout factor to both input gates and to the recurrent connections in order to prevent overfitting which is a typical issue in neural networks [104]. We have chosen a dropout factor value of 0.50.

For what concerns GxG, as we had to deal with long documents such as news, we employed a two-layer Bi-LSTM encoder. The first Bi-LSTM layer served us to encode each sentence as a token sequence, the second layer served us to encode the sequence of the sentences. For what concerns ironITA we added a task-specific Bi-LSTM for each subtask before the dense layer.

Each input word is represented by a vector which is composed by:

Word embeddings: the concatenation of the word embeddings extracted by the available Word Embedding Lexicons (128 dimensions for each word embedding), and for each word embedding an extra component was added to handle the "unknown word" (1 dimension for each lexicon used).

Word polarity: the corresponding word polarity obtained by exploiting the Sentiment Polarity Lexicons. This results in 3 components, one for each possible lexicon outcome (negative, neutral, positive) (3 dimensions). We assumed that a word not found in the lexicons has a neutral polarity.

Automatically Generated Sentiment Polarity Lexicons for Social Media: The presence or the absence of the word in a lexicon and the relative presence if the word is found in the lexicon. Since we built the TW_{GEN}, TW_{POS} and TW_{NEG} 6 dimensions are needed, 2 for each lexicon.

Coarse Grained Part-of-Speech: 13 dimensions.

End of Sentence: a component (1 dimension) indicating whether the sentence is totally read.

The K-Fold voting mechanism: Since for some tasks the training sets are relatively small and the neural-networks training procedure requires to use a portion of the training set for validation, in some tasks we used a K-Fold voting mechanism. In this setting we split the datasets in K parts and we train K classifiers using each time a different portion of the training set for validation. On testing step we activate all the classifiers on each test sample and we pick the most frequently predicted class as final prediction.

5.3.3 Results and Discussion

Table 5.6 reports the official results obtained by our best runs on all the task we participated in. As it can be noted our system performed extremely well, achieving the best scores almost

in every single subtask. In the following subsections, a discussion of the results obtained in each task is provided.

Task	Our Score	Best Score	Rank
ABSITA			
ACD	0.811	0.811	1
ACP	0.767	0.767	1
GxG IN-DOMAIN			
CH	0.640	0.640	1
DI	0.676	0.676	1
JO	0.555	0.585	2
TW	0.595	0.595	1
YT	0.555	0.555	1
GxG CROSS-DOMAIN			
CH	0.640	0.640	1
DI	0.595	0.635	2
JO	0.510	0.515	2
TW	0.609	0.609	1
YT	0.513	0.513	1
HaSpeeDe			
TW	0.799	0.799	1
FB	0.829	0.829	1
C_TW	0.699	0.699	1
C_FB	0.607	0.654	5
IronITA			
IRONY	0.730	0.730	1
SARCASM	0.516	0.520	3

TABLE 5.6: Classification results of our best runs on the ABSITA, GxG, HaSpeeDe and IronITA test sets.

5.3.4 ABSITA

We tested five learning configurations of our system based on linear SVM and DNN learning algorithms using the features described. All the experiments were aimed at testing the contribution in terms of f-score of MTL vs STL, the k-fold voting mechanism and the external resources. For what concerns the Bi-LSTM learning algorithm we tested Bi-LSTM both in the STL and MTL scenarios. In addition, to test the contribution of the Booking word embeddings, we created a configuration that uses a shallow Bi-LSTM in the MTL setting without using these embeddings (MTL NO BOOKING-WE). Finally, to test the contribution of the k-fold voting mechanism we created a configuration that does not use the k-fold voting mechanism (MTL NO K-FOLD). To obtain fair comparisons in the last case we run all the experiments 5 times and averaged the scores of the runs. To test the proposed classification models, we created an internal development set by randomly selecting documents from the training sets distributed by the task organizers. The resulting development set is composed of approximately 10% (561 documents) of the whole training set.

Configuration	ACD	ACP
baseline	0.313	0.197
linear SVM	0.797	0.739
STL	0.821	0.795
MTL	**0.824**	**0.804**
MTL NO K-FOLD	0.819	0.782
MTL NO BOOKING-WE	0.817	0.757

TABLE 5.7: Classification results (micro f-score) of the different learning models on our ABSITA development set.

Configuration	ACD	ACP
baseline	0.338	0.199
linear SVM	0.772*	0.686*
STL	**0.814**	0.765
MTL	0.811*	**0.767***
MTL NO K-FOLD	0.801	0.755
MTL NO BOOKING-WE	0.808	0.753

TABLE 5.8: Classification results (micro f-score) of the different learning models on the ABSITA official test set.

Table 5.7 reports the overall accuracies achieved by the models on the internal development set for all the tasks. In addition, the results of the baseline system (baseline row) which emits always the most probable label according to the label distributions in the training set are reported. The accuracy is calculated as the micro f–score obtained using the evaluation tool provided by the organizers. For what concerns the ACD task it is worth noting that the models based on DNN always outperform linear SVM, even though the difference in terms of f-score is small (approximately 2 f-score points). The MTL configuration was the best performing among all the models, but the difference in term of f-score among all the DNN configuration is not evident.

When analyzing the results obtained on the ACP task we can notice remarkable differences among the performances obtained by the models. Again the linear SVM was the worst performing model, but this time with a difference in terms of f-score of 6 points with respect to MTL, the best performing model on the task. It is interesting to notice that the results achieved by the DNN models have a bigger difference between them in terms of f-score with respect to the ACD task: this suggests that the external resources and the k-fold voting mechanism contribute to improve the performances in the ACP task. The configuration that does not use the k-fold voting mechanism scored 2 f-score points w.r.t. the MTL configuration. We can also notice that the Booking word embeddings were particularly helpful in this task: the MTL NO BOOOKING-WE configuration in fact scored 5 points less than the best configuration. The results obtained on the internal development set lead us to choose the models for the official runs on the provided test set. Table 5.8 reports the overall accuracies achieved by all our classifier configurations on the official test set, the official submitted runs are starred in

the table.

As it can be noticed the best scores both in the ACD and ACP tasks were obtained by the DNN models. Surprisingly the difference in terms of f-score was reduced in both the tasks, with the exception of linear SVM, which performed 4 and 8 f-score points less in the ACD and ACP tasks respectively when compared to the best DNN model systems. The STL model outperformed the MTL models the ACD task, even though the difference in term of f-score is not relevant. When the results on the ACP are considered, the MTL model outperformed all the other models, even though the difference in terms of f-score with respect to the STL model is not noticeable. Is it worth to notice that the usage of the k-fold voting mechanism and of the Booking word embeddings improves the performances of the MTL system. This can be seen by looking at the results achieved by the MTL NO BOOKING-WE model and the MTL NO K-FOLD model that scored 1.2 and 1.5 f-score points less than the MTL system.

5.3.5 GxG

We tested three different learning configurations of our system based on linear SVM and DNN learning algorithms using the features described. For what concerns the Bi-LSTM learning algorithm we tested both the STL and MTL approaches. We tested the three configurations for each of the 5 five in-domain subtasks and for each of the 5 five cross-domain subtasks. To test the proposed classification models, we created internal development sets by randomly selecting documents from the training sets distributed by the task organizers. The resulting development sets are composed of approximately 10% of each data sets. For what concerns the in-domain task, we tried to train the SVM classifier on in-domain-data only and on both in-domain and cross-domain data.

Model	**CH**	**DI**	**JO**	**TW**	**YT**
SVMa	0.667	0.626	0.485	0.582	0.611
SVM	**0.701**	0.737	0.560	0.728	0.619
STL	0.556	0.545	0.500	0.724	0.596
MTL	0.499	**0.817**	**0.625**	**0.729**	**0.632**

TABLE 5.9: Classification results of the different learning models on development set in terms of accuracy for the **in-domain** tasks.

Model	**CH**	**DI**	**JO**	**TW**	**YT**
SVM	0.530	0.565	0.580	0.588	0.568
STL	0.550	0.535	0.505	0.625	0.580
MTL	0.523	0.549	0.538	0.500	0.556

TABLE 5.10: Classification results of the different learning models on development set in terms of accuracy for the **cross-domain** tasks

Table 5.9 and 5.10 report the overall accuracy, computed as the average accuracy for the two classes (male and female), achieved by the models on the development data sets for the in-domain and the cross-domain tasks respectively. For the in-domain tasks, we observe that the

SVM performs well on the smaller datasets (Children and Diary), while MTL neural network has the best overall performances. When trained on all the datasets, in- and cross-domain, the SVM (SVMa) performs worst than when trained on in-domain data only (SVM). For what concerns the cross-domain datasets we observe poor performances over all the subtasks with all the employed models, implying that the models have difficulties in cross-domain generalization.

Model	**CH**	**DI**	**JO**	**TW**	**YT**
SVMa	0.545	0.514	0.475	0.539	**0.585**
SVM	0.550	0.649	**0.555**	0.567	0.555*
STL	0.545	0.541	0.500	**0.595***	0.512
MTL	**0.640***	**0.676***	0.470	0.561	0.546

TABLE 5.11: Classification results of the different learning models on the official test set in terms of accuracy for the **in-domain** tasks (* marks runs that outperformed all the systems that participated to the task).

Model	**CH**	**DI**	**JO**	**TW**	**YT**
SVM	0.540	0.514	**0.505**	0.586	**0.513***
STL	**0.640***	0.554	0.495	**0.609***	0.510
MTL	0.535	**0.595**	0.510	0.500	0.500

TABLE 5.12: Classification results of the different learning models on the official test set in terms of accuracy for the **cross-domain** tasks. (* marks runs that outperformed all the systems that participated to the task).

Table 5.11 and 5.12 report the overall accuracy, computed as the average accuracy for the two classes (male and female), achieved by the models on the official test sets for the in-domain and the cross-domain tasks respectively (* marks the running that obtains the best results in the competition). For what concerns the in-domain subtasks the performances appear to be not in line with the ones obtained on the development set, but still our models outperform the other participant's systems in four out of five subtasks. The MTL model provided the best results for the Children and Diary test sets, while on the other test sets all the models performed quite poorly. Again when trained on all the datasets, in and cross-domain, the SVM (SVMa) perform worst than when trained on in-domain data only (SVM). For what concerns the cross-domain subtasks, while our model gets the best performances on three out of five subtasks, the results confirm poor performances over all the subtasks, again indicating that the models have difficulties in cross-domain generalization.

5.3.6 HaSpeeDe

We tested seven learning configurations of our system based on linear SVM and DNN learning algorithms using the features described. All the experiments were aimed at testing the contribution in terms of f-score of the number of layers, MTL vs STL, the k-fold voting mechanism and the external resources. For what concerns the Bi-LSTM learning algorithm

we tested one and two layers of Bi-LSTM both in the STL and MTL scenarios. In addition, to test the contribution of the sentiment lexicon features, we created a configuration that uses a 2-layer Bi-LSTM in the MTL setting without using these features (1L MTL NO SNT). Finally, to test the contribution of the k-fold voting mechanism we created a configuration that does not use the k-fold voting mechanism (1 STL NO K-FOLD). To obtain fair results in the last case we run all the experiments 5 times and averaged the scores of the runs. To test the proposed classification models, we created two internal development sets, one for each dataset, by randomly selecting documents from the training sets distributed by the task organizers. The resulting development sets are composed by 10% (300 documents) of the whole training sets.

Configuration	**TW**	**FB**	**C_TW**	**C_FB**
baseline	0.378	0.345	0.345	0.378
linear SVM	**0.800**	0.813	0.617	0.503
1L STL	0.774	**0.860**	0.683	0.647
2L STL	0.790	**0.860**	0.672	0.597
1L MTL	0.783	**0.860**	0.672	**0.663**
2L MTL	0.796	0.853	**0.710**	0.613
1L MTL NO SNT	0.793	0.857	0.651	0.661
1L STL NO K-FOLD	0.771	0.846	0.657	0.646

TABLE 5.13: Classification results of the different learning models on our HaSpeeDe development set in terms of F1-score.

Configuration	**TW**	**FB**	**C_TW**	**C_FB**
baseline	0.403	0.404	0.404	0.403
best official system	0.799	0.829	0.699	0.654
linear SVM	0.798*	0.761	0.658	0.451
1L STL	0.793	0.811*	0.669*	0.607*
2L STL	0.791	0.812	0.644	0.561
1L MTL	0.788	0.818	**0.707**	**0.635**
2L MTL	0.799*	**0.829***	0.699*	0.585*
1L MTL NO SNT	**0.801**	0.808	0.709	0.620
1L STL NO FOLD	0.785	0.806	0.652	0.583

TABLE 5.14: Classification results of the different learning models on the official HaSpeeDe test set in terms of F1-score.

Table 5.13 reports the overall accuracies achieved by the models on our internal development sets for all the tasks. In addition, the results of the baseline system (baseline row) which emits always the most probable label according to the label distribution in the training set is reported. The accuracy is calculated as the f–score obtained using the evaluation tool provided by the organizers. For what concerns the Twitter in–domain task (TW in the table) it is worth noting that linear SVM outperformed all the configurations based on Bi-LSTM. In addition, the MTL architecture results are slightly better than the STL ones (+1 f-score point

with respect to the STL counterparts). External sentiment resources were not particularly helpful in this task, as shown by the result obtained by the 1L MTL NO SNT row. In the FB task, Bi-LSTMs sensibly outperformed linear SVMs (+5 f-score points on average); this is most probably due to longer text lengths that are found in this dataset with respect to the Twitter one. For what concerns the out–domain tasks, when testing models trained on Twitter and tested on Facebook (C_TW column), we can notice an expected drop in performance with respect to the models trained on the FB dataset (15-20 points f-score points). The best result was achieved by the 2L MTL configuration (+4 points w.r.t. the STL counterpart). Finally, when testing the models trained on Facebook and tested on Twitter (C_FB column), linear SVM showed a huge drop in terms of accuracy (-30 f-score points), while all the models trained with Bi-LSTM showed a performance drop of approximately 12 f-score points. Also in this setting, the best result was achieved by an MTL configuration (1L MTL), which performed better with respect to the STL counterpart (+2 f-score points). For what concerns the k-fold voting mechanism, we can notice that the results achieved by the model not using the k-fold learning strategy (1 STL NO K-FOLD) are always lower than the counterpart which used the k-fold approach (+2.5 f-score points gained in the C_TW task), showing the benefits of using this technique.

These results lead us to choose the models for the official runs on the provided test set. Table 5.14 reports the overall accuracies achieved by all our classifier configurations on the official test set, the official submitted runs are starred in the table. The *best official system* row reports, for each task, the best official results submitted by the participants of the EVALITA 2018 HaSpeeDe shared task. As we can note the best scores in each task were obtained by the Bi-LSTM in the MTL setting, showing that MTL networks seem to be more effective with respect to STL networks. For what concerns the Twitter in–domain task, we obtained similar results to the development set ones. A sensible drop in performance is observed in the FB task w.r.t the development set (-5 f-score points on average). Still, Bi-LSTMs models outperformed the linear SVM model by 5 f-score points. In the cross-domain tasks, all the models performed similarly to what observed in the development set. It is worth observing that linear SVM performed almost like a baseline system in the C_FB task. In addition, in the same task, the model exploiting the sentiment lexicon (1L MTL) showed better performance (+1.5 f-score points) w.r.t to the 1L MTL NO SNT model. It is worth noticing that the k-fold voting mechanism was beneficial also on the official test set: the 1L STL model obtained better results (approximately +2 f-score points in each task) w.r.t. the the model that did not use the k-fold voting mechanism.

5.3.7 IronITA

We tested four learning configurations of our system. The first one is based on linear SVM. The other three are based on deep neural network (DNN) and MTL. The first neural setting (MTL) is trained on the IronITA training dataset (exploiting both the label of subatask A and B). The second neural setting (MTL+Polarity) is trained on IronITA and SENTIPOLC polarity [11] datasets. The third neural setting (MTL+Polarity+Hate) is trained on IronITA,

SENTIPOLC and HaSpeeDe datasets. To select the proposed classification models, we used k-cross validation (k=4).

Configuration	**Irony**	**Sarcasm**
linear SVM	0.734	0.512
MTL	0.745	0.530
MTL+Polarity	0.757	**0.562**
MTL+Polarity+Hate	**0.760**	0.557

TABLE 5.15: Classification results of the different learning models on k-cross validation terms of average F1-score.

Configuration	**Irony**	**Sarcasm**
baseline-random	0.505	0.337
baseline-mfc	0.334	0.223
best participant	0.730	0.52
linear SVM	0.701	0.493
MTL	**0.736**	**0.530**
MTL+Polarity	0.730*	0.516*
MTL+Polarity+Hate	0.713*	0.503*

TABLE 5.16: Classification results of the different learning models on the official test set in terms of F1-score (* submitted run).

Table 5.15 reports the overall average f-score achieved by the models on the k-cross validation sets for both the irony and sarcasm detection tasks.

We can observe that the SVM obtains good results on irony detection but the MTL neural approach overperforms sensibly the SVM. Also, we note that the usage of additional Polarity and Hate Speech datasets lead to better performances. These results lead us to choose the MTL models trained with the additional datasets for the two official run submissions.

Table 5.16 reports the overall accuracies achieved by all our classifier configurations on the official test set, the official submitted runs are starred in the table. The accuracies have been computed in terms of F-Score using the official evaluation script. We submitted the runs MTL+Polarity and MTL+Polarity+Hate. The run MTL+Polarity ranked first and third in subtask A and B respectively on the official leaderboard. The run MTL+Polarity ranked second and fourth in subtask A and B on the official leaderboard.

The results on the test set confirm the good performances of the SVM classifier on the irony detection task and that the MTL neural approaches overperform the SVM. The model trained on the IronITA and SENTIPOLC datasets outperformed all the systems that participated in subtask A, while on the subtask B it slightly underperformed the best participant system. The model trained on the IronITA, SENTIPOLC and HaSpeeDe datasets overperformed all the systems that participated in subtask A but our model trained on IronITA and SENTIPOLC datasets only. Although the best scores in both tasks were obtained by the MTL network trained on the IronITA data set only. The MTL model trained on the IronITA dataset only

would have outperformed all the systems submitted to both the subtasks by all participants. Seems that for these tasks the usage of additional datasets leads to overfitting issues.

5.3.8 Conclusions

In this section, we reported the results of our participation in the ABSITA, GxG, HaSpeeDe and IronITA shared tasks of the EVALITA 2018 conference. By resorting to a system that used Support Vector Machines and Deep Neural Networks (DNN) as learning algorithms, we achieved the best scores almost in every task, showing the effectiveness of our approach. In addition, when DNN was used as a learning algorithm we introduced a new multi-task learning approach and a majority vote classification approach to further improve the overall accuracy of our system. The proposed system resulted in a very effective solution achieving the first position in almost all sub-tasks for each shared task. Almost in all subtasks, neural approaches led to better performances than the SVM classifiers. However, for the GxG task, the accuracy scores are really low showing that for author profiling further works are needed. Another important aspect is that in cross-domain scenarios (HaSpeeDe and GxG) the systems obtained low scores indicating that the described approaches are not robust enough to deal with domain switching.

5.4 More on author profiling: EVALITA 2020

5.4.1 Introduction

In the context of EVALITA 2020 [20], the periodic evaluation campaign of Natural Language Processing and speech tools for the Italian language, the task TAG-it [57] has been proposed. TAG-it is an Author Profiling task in which the goal is to provide a system capable of predicting the gender and the age of the authors of several blog posts and their topics. This task can be considered as a follow-up of the EVALITA 2018's GxG. In GxG the results obtained are lower than ones observed in other campaigns and languages. In order to address this problem and get better performances, in TAG-it only blogs' genre is considered and longer texts are used since they provide more evidence than tweets and Youtube comments, which are shorter than blog posts. Moreover, with respect to GxG, TAG-it adds the topic control with the aim of evaluating the interaction of topic and lexically rich models on performances in a more direct way than in GxG, in which this was indirectly done via cross-genre prediction. TAG-it is divided in two subtasks: the goal of the first one (Subtask 1) is to classify gender, age and topic at once, while the goal of the second one is to predict age (Subtask 2a) and gender (Subtask 2b) separately with topic control.

The previous works described in this chapter demonstrated the validity of the Multi-Task Learning approach to establish the state of the art for several Italian NLP task, in the context of GxG, the presented system obtained the best results. For TAG-it we replicated the same approach: we developed a baseline system based on SVM, and two neural systems, the first one exploiting a Single-Task Learning approach, the second one a Multi-Task Learning

approach. Instead of the Bi-LSTM model for TAG-it we exploited a deeper neural pre-trained language model: BERT [86].

5.4.2 Description of the Systems

We implemented and tested three different systems. Our early experiments were led on a training set and a test set obtained by shuffling and splitting (80% training - 20% test) the training set provided by the organisers in order to analyse the classifiers' performances on a labelled dataset. At the end of our experiments, we trained our best classifiers on the whole training set and run them on the TAG-it test sets provided by the organisers.

For our experiments and runs, as a preprocessing phase, we filtered out all posts less than 20 characters in length and labelled each post of the dataset with the corresponding author's id, gender, age and topic. In Table 5.17 we report the distributions of the classes of the TAG-it dataset.

	Train	**Test1**	**Test2a**	**Test2b**
M	15070	315	344	730
F	3113	96	68	69
0-19	2232	39	76	79
20-29	5412	131	189	230
30-39	3569	95	51	134
40-49	3577	69	48	216
50-100	3393	77	48	140
ANIME	3925	97	0	0
AUTO-MOTO	3648	76	0	0
BIKES	468	12	0	0
CELEBRITIES	1063	22	0	0
ENTERTAINMENT	534	9	0	0
MEDICINE-AESTHETICS	370	16	0	0
METAL-DETECTING	1471	26	0	0
NATURE	481	11	0	0
SMOKE	1574	30	0	0
SPORTS	4593	103	0	0
TECHNOLOGY	56	9	0	0
GAMES	0	0	298	298
ROLE-GAMES	0	0	114	114
CLOCKS	0	0	0	387

TABLE 5.17: TAG-it datasets distributions

As a first step, our systems make their predictions by classifying the three dimensions post by post. Then they use a voting mechanism according to which the gender, age and topic of an author are represented by the most frequent values assigned by the classifiers to his/her posts.

The first system we implemented uses linear Support Vector Machine as a learning algorithm and we used different features for predicting the core dimensions of the dataset, the second system is based on a Single-Task Learning BERT model and the third system is

based on a Multi-Task Learning BERT model. In particular, we used UmBERTo[7], an Italian pretrained Language Model developed by Musixmatch.

In the following subsections, we will describe these systems in detail.

5.4.3 Support Vector Machine Classifiers

As regards the system based on three linear SVM statistical models, we used the scikit-learn [233] Python library and we conducted several experiments by testing different configurations for feature extraction. In all the experiments we used the TF-IDF vectorizer, but we changed the tokenizer and the *n*-grams context window. In particular, we tested five different kinds of features: character *n*-grams, word *n*-grams, lemma *n*-grams, Part-Of-Speech *n*-grams and bleached tokens. As regards the bleached tokens features, they were extracted after performing bleach tokenization consisting in fading out lexicon in favour of an abstract token representation [118]. The word *n*-grams, lemma *n*-grams and Part-Of-Speech *n*-grams features were extracted by using the linguistic pipeline for the Italian language provided by spaCy[8]. For the multi-class classification we applied the One-Vs-Rest method [256]. In Table 5.18 we report the performances in terms of micro-average f-score of the SVM models tested in our experiments.

These results led us to choose the best SVM classifiers for the official runs on the provided test set; analysing them, we can state that the best SVM classifiers tested in our experiments are the following:

- Topic Detection: One-Vs-Rest Linear SVM using features extracted through a TF-IDF Vectorizer considering character *n*-grams;
- Age Detection: One-Vs-Rest Linear SVM using features extracted through a TF-IDF Vectorizer considering lemma *n*-grams;
- Gender Detection: Linear SVM using features extracted through a TF-IDF Vectorizer considering word *n*-grams.

	Gender	**Age**	**Topic**
word n-gram	**0.933**	0.3873	0.7882
char n-gram	0.9284	0.3739	**0.8333**
lemma n-gram	0.9265	**0.4189**	0.7928
pos n-gram	0.9223	0.3063	0.3873
bleached words	0.9223	0.3739	0.4775

TABLE 5.18: SVM classifiers' micro-average f1-scores on validation set

[7] https://github.com/musixmatchresearch/umberto
[8] https://spacy.io

5.4.4 Single-Task BERT-based Classifiers

Our second system consists of three different BERT models and a classifier on top of each of them. More precisely, we used the UmBERTo language model, which was pretrained on a large Italian Corpus: OSCAR [230].

This language model has 12-layer, 768-hidden, 12-heads, 110M parameters. On top of the language model, we added a ReLU classifier [219]. We applied dropout [283] to prevent overfitting. As loss function, we used the sum of loss functions of the three classifiers. For each classifier, we used Cross-Entropy as a loss function.

In Table 5.19 we report the system's performances in terms of f1-score obtained on the validation set.

	f1-score
Gender	0.86
Age	0.35
Topic	0.66

TABLE 5.19: Single-Task Learning BERT-based system micro-average f1-scores on validation set

5.4.5 Multi-task BERT-based Classifier

Our last system is based on a unique UmBERTo model and three classifiers on top of it, each one responsible for predicting one of the three core dimensions of the dataset according to the Multi-Task Learning approach used in [58]. On top of the model we added three ReLU classifiers, we applied the dropout method and we used the sum of the Cross-Entropy loss functions of the three classifiers as loss function.

In Table 5.20 we report the system's performances in terms of f1-score obtained on the validation set.

	f1-score
Gender	0.86
Age	0.39
Topic	0.64

TABLE 5.20: Multi-Task Learning BERT-based system f1-scores on validation set

5.4.6 Results and Evaluation

We run all our three systems on the test sets provided by the task organisers. The performances of our systems are reported in Table 5.21.

For Task 1 scoring, TAG-it considers two different rankings. The first ranking is obtained using a partial scoring scheme, giving 0 points if no correct predictions are provided for the three dimensions of the dataset, 1/3 points if one out of three correct answers is given, 2/3 points if two out of three correct answers are given and 1 point if all the answers given by the

system are correct. The second-ranking assigns 0 points if no correct predictions are provided for the three dimensions of the dataset and 1 point if all the answers given by the system are correct. In both cases, the final score is the sum of the points achieved by the system across all the documents normalized with respect to the number of documents in the test set. For Task 2, the micro-average f-score is used as a scoring function.

	STL-SVM	**MTL-BERT**	**STL-BERT**
Task 1 metric 1	0.6626	0.7178	**0,7348**
Task 1 metric 2	0.253	0.3090	**0,3309**
Task 2a	0.8519	**0.9247**	0,9053
Task 2b	0.3742	0.3667	**0,4093**

TABLE 5.21: Systems' performances evaluation with TAG-it metrics

Analysing the scores in Table 5.21, we can state that the best system in the TAG-it context is the one based on BERT using the Single-Task Learning (STL-BERT) approach, obtaining the best scores in Task 1 and Task 2b (age prediction). In Task 2a, consisting of gender prediction with topic control, the best system is the Multi-Task Learning BERT-based system (MTL-BERT). Hence, the systems based on deeper neural models outperform the systems based on traditional machine learning techniques, i.e. the SVM (STL-SVM).

Task 1: In order to compare classifiers' predictions on Task 1 with regard to each dimension and to understand the correlation between labels, we plotted and analysed some distributions.

In Figure 5.2, we reported the distribution of the labels in the test set and in the classifiers' output. As regards the gender prediction (a), we can note that the STL-SVM classifier overestimates the `M` class, most likely because the `M` and `F` classes are very unbalanced in the training set. STL-BERT and MTL-BERT's distributions, on the contrary, are closer to the test set's one: in our setting the neural models appear less affected by the imbalance of a training set.

Observing the distributions of the Age classes in Figure 5.2 (b), we can observe that for all three systems the distributions of the labels are not close to the distribution of the test set. The nearest distribution is one of MTL-BERT's output.

Looking at the Topic classes distributions in Figure 5.2 (c), we can observe, once again, that the SVM-based system's one is the less close to the test set in that it has the tendency to overestimate the `SPORT`, `ANIME` and `AUTO-MOTO` classes and it does not recognise the `BIKES` and `TECHNOLOGY` classes as they are underrepresented in the training set (respectively the 2.574% and the 0.308% of training set). For the same reason, it has difficulties in recognising the classes `ENTERTAINMENT`, `MEDICINE-AESTHETICS` and `NATURE` (which are respectively the 2.937%, 2.035% and 2.645% of the training set). The two BERT-based systems, on the contrary, are less affected by this imbalance of the training set and their predictions reflect more the reality of the test set, even though, as STL-SVM, also MTL-BERT cannot recognise the `BIKES` and `TECHNOLOGY` classes.

In Figure 5.3 we report the distribution of the Age classes with respect to the Topic classes. Figure 5.3 (b) shows that in the STL-SVM's output the `0-19` age class is only

related to the ANIME topic, the age 20-29 is related more or less with all the detected topics, the 30-39 class is mostly related to SMOKE and MEDICINE-AESTHETICS, the 40-49 class to the METAL-DETECTING, AUTO-MOTO and SMOKE topics and the 50-100 class mostly to AUTO-MOTO, SPORTS and CELEBRITIES. This distribution is quite far from the test set one and it seems that the relation between the class 0-19 and the topics is overestimated. In Figure 5.3 (c), which refers to MTL-BERT, we can note that authors classified as having age 20-29 are predicted to talk mostly about ANIME, CELEBRITIES, NATURE and SPORTS and are less related to ENTERTAINMENT, MEDICINE-AESTHETICS and NATURE topics than in STL-SVM's output; the relation between the 30-39 class and ENTERTAINMENT and MEDICINE-AESTHETICS categories on one hand, and 50-100 and AUTO-MOTO, MEDICINE-AESTHETICS, METAL-DETECTING, NATURE and SMOKE on the other is stronger than in STL-SVM's results. Also this distribution, though, is quite far from the test set's one, even if ages seem to be more distributed than in STL-SVM's output. As shown in Figure 5.3 (d), in STL-BERT's distribution, the age 0-19 seems mostly related to TECHNOLOGY and ANIME. The class BIKES, which has not been recognised by the other systems, is related to the classes 30-39, 40-49 and, mostly, 50-100. As regards the 20-29 class, its relations are quite similar to the ones found in the STL-SVM's results, except for the class NATURE, which is related also to the ages 0-19, 40-49 and 50-100. Also this distribution is quite far from the test's one. All the three distributions differ considerably from the test set because systems do not perform well enough in age prediction.

The distributions of the topics with respect to gender in the test set and the predictions are reported in Figure 5.4. As shown in the figure, all the three systems results relate the F class mostly to the ANIME topic, as it is also in the test set. In the STL-SVM's output, though, this relation seems to be overestimated. Moreover, in STL-SVM the F class, besides ANIME, is only related to a much lesser extent to SMOKE. The relation between M and SMOKE seems to be overestimated too with respect to the test set. As regards the F class in MLT-BERT and STL-BERT outputs, topics are more distributed than in STL-SVM, but the nearest to the test set's one is STL-BERT: MLT-BERT, in fact, seems to overestimate the relation between F and BIKES and ENTERTAINMENT and to underestimate the relation between F and MEDICINE-AESTHETIC and SPORTS. For what concerns the M class in MLT-BERT and STL-BERT distributions, we can state once again that the distribution which is closer to the test set one is given by STL-BERT: STL-SVM, MLT-BERT overestimates the relation between M and SMOKE and NATURE.

Task 2: The results reported in Table 5.21 show that for Task 2a (gender prediction with topic control) the best classifier is MLT-BERT. In this subtask, BERT-based systems outperform in a significant way the system based on SVM.

As regards the Task 2b, consisting of the age prediction, the best metrics belong to the STL-BERT. In the age prediction, the gap between all the systems' metrics is not very high. In this case, in which only the age dimension must be predicted, the best classifier is the one using a Single-Task Learning approach.

5.4.7 Conclusions

In this work, we reported the performances and the results of the systems we used to participate in the TAG-it task of EVALITA 2020. We compared our systems' performances and noted that in the case in which the goal is to predict topic, age and gender dimensions at once, and in the case in which only the age must be predicted, the best classifier is the one developed using a Single-Task Learning approach and based on transformers. In the case in which the goal is the gender prediction only a Multi-task Learning approach combined with transformers has slightly better performances. These results prove that the proposed systems based on transformers are more effective than traditional machine learning techniques in the topic, age and gender classification achieving the state of the art for TAG-it shared task. Using deep pre-trained language models on this task Multi-Task Learning does not provide any relevant boost of performances. As mentioned, TAG-it could be seen as a continuation of the GxG task at EVALITA 2018. In the latter, teams were asked to predict gender within and across five different genres. We observe that results at TAG-it for gender prediction are higher than in GxG both within and cross-domain. This is might be ascribed to three main factors: (i) in this editions authors were represented by multiple texts, while in GxG, for some domains, evidence per author was minimal; (ii) texts in TAG-it are probably less noisy, at least in comparison to some of the GxG genres (e.g., tweets and YouTube comments); (iii) transformer-based model (which were not widely available in 2018) provided a boost of performances.

5.5 Final Remarks

In this section it has been reported the work done to build models able to model affect aspects such as sentiment, hate speech and irony. Also, stylistic aspects related to author profiling have been modelled. The systems developed represent the state of the art for Italian for each task in which they have been tested. Turns out that is possible to analyze whit a good level of accuracy those aspects. However, none of the mentioned systems has been applied without a drop in performances in cross-domain scenarios, suggesting that there is a need to work on these models to make them more robust. However those model are able to capture specific variational aspects, in the next chapters, we will investigate how those approaches can be used to evaluate the ability to replicate the same variational aspects by NLG systems.

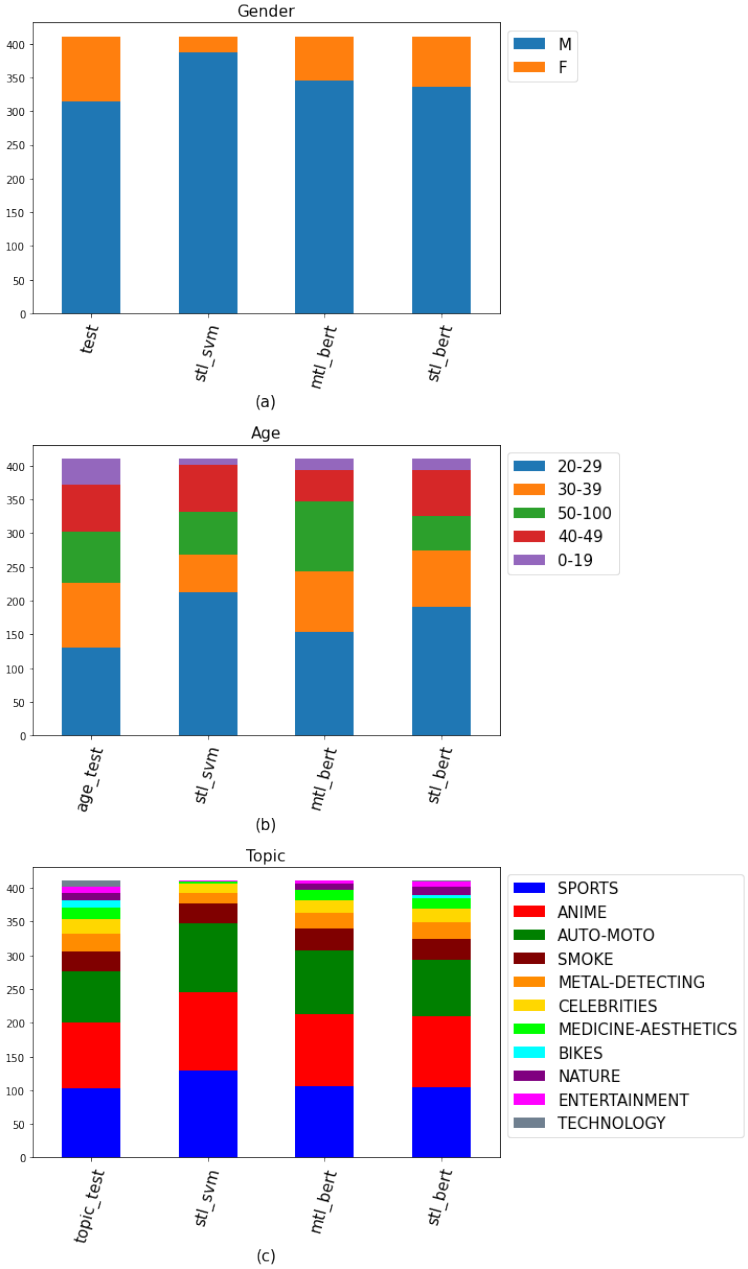


FIGURE 5.2: Task 1, Distributions of the dimensions' classes in test set and classifiers' predictions.

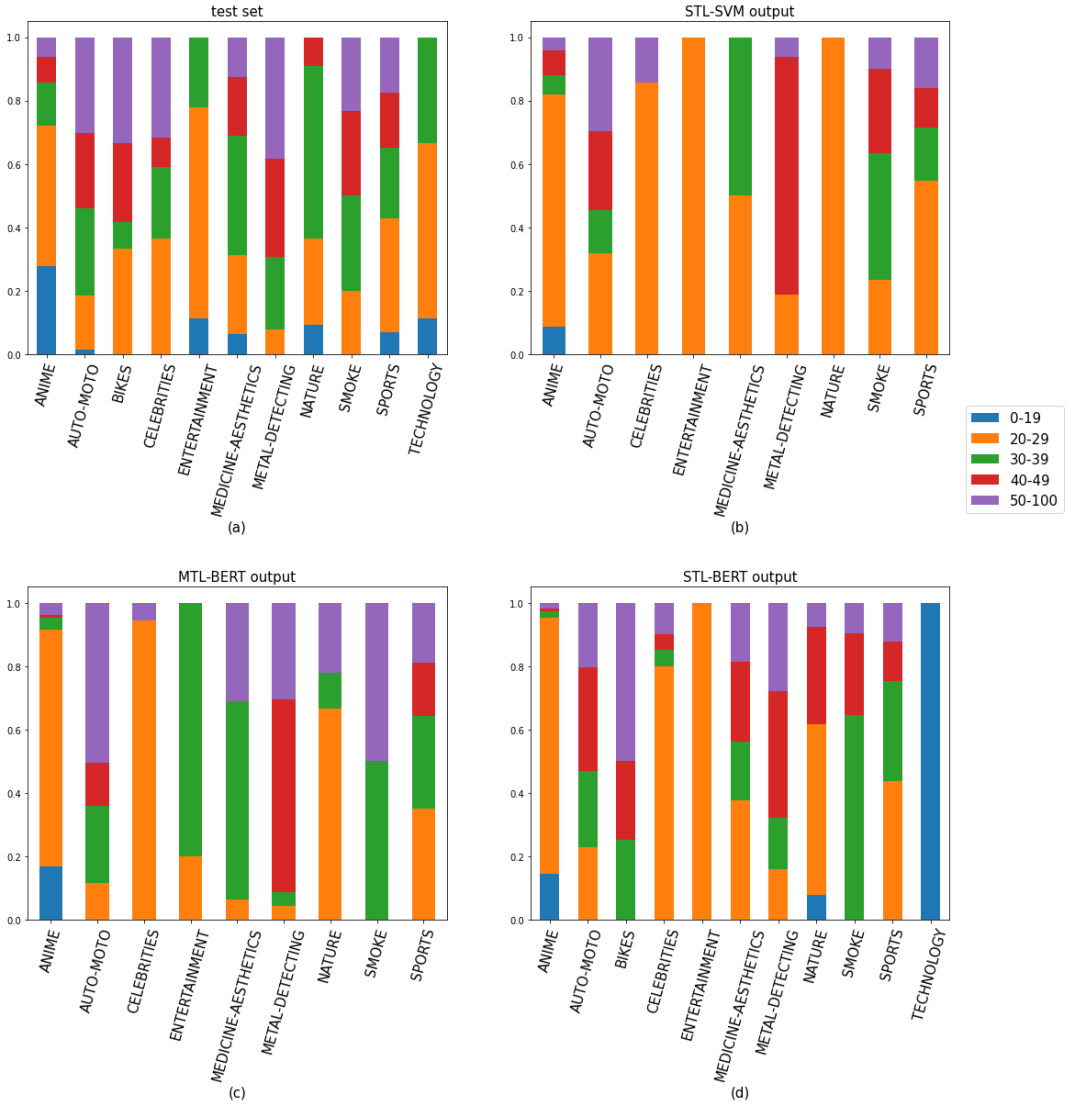


FIGURE 5.3: Task 1, Distributions of the Topic and Age dimensions in test set and classifiers' predictions.

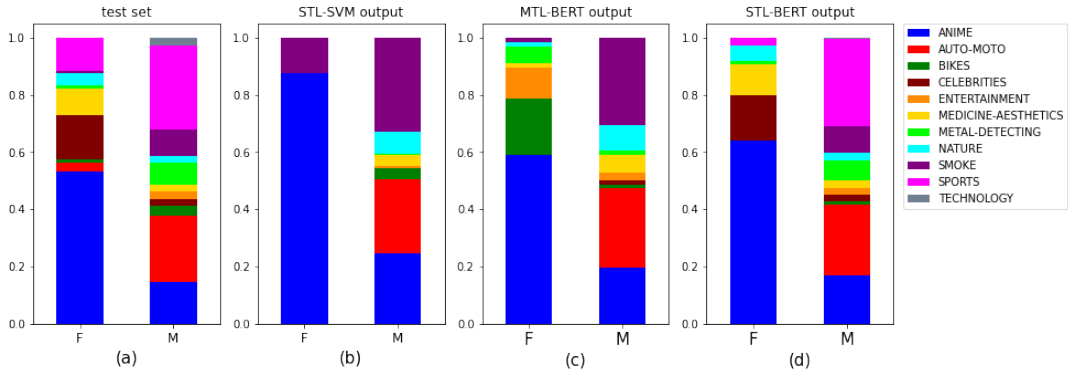


FIGURE 5.4: Task 1, Distributions of the Topic and Gender dimensions in test set and classifiers' predictions.

Chapter 6

Polarization in News Headline

6.1 Introduction

Different newspapers especially if positioned at opposite ends of the political spectrum, can render the same event in different ways. Newspaper-specific style is likely to be exhibited not only in the articles' body but also in the headlines, which are a prime tool to capture attention and make clear statements about the newspaper's position over a certain event. This context provides us with an excellent scenario to test the capability of NLU models to capture the stylistic variation of the two newspapers taken into consideration and the capabilities of the NLG system to reproduce such stylistic variations. In this chapter, several studies done on this topic are reported. Firstly we trained three sequence to sequence model to generate news headlines and we exploited human evaluation to assess the quality of these models considering a few different aspects independently by the newspaper-specific styles. Then we used word embedding shifts to analyze different word use in the two newspaper. Then we focused on style by developing a system to generate headlines accordingly to the style of the two newspapers and we investigated how NLU techniques can be exploited to assess the NLG systems ability to learn stylistic aspects. Finally, we framed a style transfer task for the news headline of the two newspapers and we tested two different approaches to tackle this task. These tasks consider both the capability of NLG systems in reproducing the target style and their ability in preserving contents. The task has been proposed at EVALITA 2020 as the first NLG shared task ever presented in the EVALITA campaigns.

6.2 Suitable Doesn't Mean Attractive. Human-Based Evaluation of Automatically Generated Headlines

Progress in language generation has made it really hard to tell if a text is written by a human or is machine-generated. But what makes generated text *good* text? In [40] we investigated this question in the context of automatically generated news headlines.[1]

Headlines could be seen as very short summaries so that one could use evaluation methods typical of summarisation [105], but they are in fact a very special kind of summaries. In

[1]A growing interest in headline generation is witnessed also in the organisation of a multilingual shared task at RANLP 2019, using Wikipedia data: http://multiling.iit.demokritos.gr/pages/view/1651/task-headline-generation

addition to being suitable in terms of content, newspaper titles must also be inviting towards reading the whole article. A model that, given an article, learns how to generate its title must then be able to cover both the summarization as well as the luring aspect.

In contrast to the feature-rich approach of [68], which requires substantial linguistic preprocessing for feature extraction, we rely on neural architectures and train three different sequence-to-sequence models that learn to generate a headline given (a portion of) its article. We compare these generated headlines to one another and to the gold headline through a series of human-based evaluations which take several aspects into account, ranging from grammatical correctness to attractiveness towards reading the full article. The factors we measure are in line with the requirements for human-based evaluation mentioned by [105], and are useful since it is known that standard metrics based on lexical overlap are not accurate indicators for the goodness of generated text [182].

Contributions We offer three main contributions: (i) a model which generates headlines from Italian news articles and which we make publicly available; (ii) a framework for human-based evaluation of generated headlines, which can serve as a blueprint for the evaluation of other types of generated texts; (iii) insights on the performance of different headline generators, and on the distinction between the concepts of suitable and attractive when evaluating headlines.

model	example generated headlines
`s2s`	Al Qaida : " L' Europa non è un pericolo per i nostri fratelli " la Samp batte la Sampdoria e la Samp non si ferma mai
`pn`	Teramo , bimbo di sei anni muore sotto gli occhi dei genitori mentre faceva il bagno Brescia , boa constrictor : sequestrati due metri e mezzo in un anno di animali
`pnc`	Argentina , Obama : " Paladino dei poveri e dei piu vulnerabili " . E il Papa si divide Cagliari , cane ha preferito rimandare il cane dal veterinario di Santa Margherita di famiglia

TABLE 6.1: Examples of headlines generated by the three models.

6.2.1 Task, Data, and Settings

The task is conceptually straightforward: given an article, generate its headline. Luckily, correspondingly straightforward is obtaining training and test data. We scraped the websites of two major Italian newspapers, namely *La Repubblica*[2] and *Il Giornale*[3], collecting a total of approximately 275,000 article-headline pairs. The two newspapers are not equally represented, with *Il Giornale* covering 70% of the data.

After removing some duplicates, and instances featuring headlines shorter than 20 characters (which are typically commercials), we were left with a total of 253,543 pairs, which we split into training (177,480), validation (50,709), and test (25,354) sets, preserving in each the proportion of the two newspapers.

[2] https://www.repubblica.it
[3] http://www.ilgiornale.it

We used the training and validation sets to develop three different models that learn to generate a headline given an article. To keep training computationally manageable, each article was truncated after the first 500 tokens.[4] As an alternative to keep the text short but maximally informative, we also experimented with selecting relevant portions of the articles using the TextRank algorithm, a graph-model that ranks sentences in a text according to their importance [208]. However, preliminary experiments on our validation set did not seem to yield better results than just selecting the first N-tokens of an article. Also, using TextRank would make a less natural comparison to the settings used for the human evaluation (see Section 6.5.3), so we did not pursue this option further.

6.2.2 Models

The models that we trained and evaluated are described below. In Table 6.1 we show two generated examples for each of the three models to give an idea of their output.

Sequence-to-Sequence with Attention (S2S) We used a sequence-to-sequence model [288] with attention [8] with the configuration used by [272] but we used a bidirectional instead of a unidirectional layer. This choice applies to all the models we used. The final configuration is 1 bidirectional encoder-decoder layer with 256 LSTM cells each, no dropout and shared embeddings with size 128; the model is optimised with Adagrad with learning rate 0.15 and gradient clipped [209] to a maximum magnitude of 2. We experimented also with a version using pretrained Italian embeddings, but since some preliminary evaluation didn't show better results, we eventually decided not to use this other model.

Pointer Generator Network (PN) The hybrid pointer-generator network architecture [272] can copy words from the source text via a *pointing mechanism*, and generate words from a fixed vocabulary. This allows for better handling of out-of-vocabulary words, providing accurate reproduction information while retaining the ability to reproduce novel words. The base architecture is a sequence-to-sequence model, except for the pointing mechanism and for the fact that the copy attention parameters are shared with regular attention. An additional layer (so-called *bridge* [159]) is trained between the encoder and the decoder and is fed with the latest encoder states. Its purpose is to learn to generate initial states for the decoder instead of initialising them directly with the latest encoder states.

Pointer Generator Network with Coverage (PNC) This model is basically a Pointer Generator Network with an additional coverage attention mechanism that is intended to overcome the copying problem typical of sequence-to-sequence models [272]. This is basically a vector, computed by summing up all the attention distributions over all previous decoder timesteps. This unnormalised distribution over the document words is expected to represent the degree of coverage that the words have received from the attention mechanism until then. This vector, called *coverage vector*, is used to penalise the attention over already generated words, to minimise the risk of generating repetitive text.

[4] We do not control for sentence endings, so the last sentence of each truncated article might get truncated.

6.2.3 Evaluation

Evaluating an automatically generated text is non-trivial. Given that many different generated texts can be correct, existing measures are usually deemed insufficient [182]. The problem is even more acute for headline generation since due to their nature and function, simple content evaluation based on word overlap is most likely not exhaustive. Human-based evaluation could provide a richer picture.

When discussing human-based (intrinsic) evaluation of summarization models, [105] mention two core aspects: *linguistic fluency or correctness*, and *adequacy or correctness relative to the input*, in terms of the system's rendition of the content. These also relate to the aspects examined in the context of evaluating the generation of the final sentence of a story, such as *grammaticality*, *(logical) consistency*, and *context relevance* [180].

We took these factors into consideration when designing our evaluation settings. Since headlines must also carry some "attraction" factor to read the whole article, we included this aspect as well.

Settings

We call a case each set of an article and its four corresponding headlines to be evaluated, namely the three automatically generated ones, and the original (gold) title.

We prepared an evaluation form[5], which included five different questions for each case (see Figure 6.1). Each subject could see the four headlines and answer questions Q1–Q3. The corresponding article, in the truncated form that was also seen in training by the models, was only shown to the subjects after Q3, and they would then answer Q4–Q5. This choice was made in order to ensure that the first questions were answered on the basis of the headlines only, especially for the validity of Q3. The order in which gold and generated titles were shown was randomised, though it was the same for each case for all participants.

Each form comprised 20 cases to evaluate and was sent to 3 participants. We created 10 different forms, thus obtaining judgements for 200 total cases with 30 different participants (600 separate judgements). The participants are all native speakers of Italian and balanced for gender (15F/15M). We also aimed at a wide range of ages (17–77) and education levels (middle school diploma to PhD). This variety was sought in order to prevent as much as possible judgements that are based too strongly on personal biases, taste, and familiarity with specific topics over others.

The headlines used for this evaluation exercise were randomly selected from the test set. When extracting them though, we excluded all cases where at least one model produced a headline containing at least an unknown word (represented with the special token $< UNK >$), since this would make the headline look too weird and not much comprehensible. This led to excluding approximately 50% of the samples. The model with the highest proportion of headlines with at least one UNK was the S2S (37%), followed by the PNC (31%), and the PN (30.2%). In terms of topics, random picking ensured a variety of topics; manual inspection anyway showed that most news was mainly about chronicle facts and international politics.

[5] An example can be found here: https://forms.gle/MB31uEGT856af2MP7

The four titles are shown (repeated for each question below)

A. Usa , la fabbrica del vetro d' aria per il telefono d' aria in Usa
B. Se il lavoro va ai robot : un automa vale sei operai
C. Usa , Trump : " Trump si difende l' occupazione e l' economia nazionale "
D. Usa , la beffa del condizionatore d' aria " made in Usa " : " Ecco come si difende "

And the following questions are then asked:

[at this stage the subjects only see titles, without the article]	
Q1. Questi titoli sono scritti correttamente?	yes,no for each
Q2. Secondo te, questi titoli parlano dello stesso articolo?	yes,no for pairs of titles
Q3. Quale di questi titoli ti invoglia maggiormente a leggere l'intero articolo?	pick one
[now the subjects also see the (truncated) article]	
New York . Chiamiamola la beffa del condizionatore d' aria " made in Usa " . La marca è Carrier , filiale della multinazionale United Technologies . Un caso ormai celebre , che Donald Trump addita come un esempio della sua azione efficace a tutela della classe operaia . A novembre , appena eletto presidente (ma non ancora in carica) , Trump si occupa dello " scandalo Carrier " : vogliono chiudere una fabbrica di condizionatori a Indianapolis per trasferirla in Messico , delocalizzando a Sud del confine 800 posti di lavoro . Il presidente - eletto fa fuoco e fiamme , chiama il chief executive dell' azienda . Forse interviene la casa madre , United Technologies , che ha grosse commesse per l' esercito e non vuole inimicarsi il neo - presidente . Sta di fatto che Carrier cede alle pressioni , fa dietrofront : la fabbrica resta sul suolo Usa , nello Stato dell' Indiana . Tripudio di Trump che canta vittoria via Twitter : " Ecco come si difende l' occupazione e l' economia nazionale " . Passano i mesi e il caso viene dimenticato . Fino a quando il chief executive Greg Hayes rivela ai sindacati che i 16 milioni di investimento nella sede di Indianapolis vanno tutti in robotica , automazione : " Alla fine ci saranno meno posti di prima . Dobbiamo ridurre i costi , per essere competitivi " . La morale è crudele , la vittoria di Trump si [. . .]	
Q4. Ritieni che il titolo sia appropriato all'articolo?	yes,no for each
Q5. Quale ti sembra più adatto? Ordinali	rank 1–4

FIGURE 6.1: Sample evaluation case. Subjects are presented with the gold and generated headlines in random order, and must answer a progression of questions, without and with seeing the article. Q1 targets correctness, Q2 targets the similarity in topic focus, Q3 targets attractiveness, Q4 and Q5 target appropriateness (absolute, and relative to one another). In this example, A=s2s, B=gold, C=pnc, D=pn.

Analysis

We discuss the results in detail for questions Q1, Q3, Q4, Q5. For Q2, we simply note that the most similar in content are always the two pointer networks, and the most dissimilar are all three pairs that involve the gold headlines. This suggests that human titles focus on aspects of the article that are different from those picked by the generator, most likely as humans can abstract away from the actual text and use much more creativity.

Grammatical Correctness (Q1) When asked to evaluate whether the headlines were written correctly, the participants assessed all headlines as correct more frequently than not correct, with Gold and PN having the best ratio of yes vs no (Figure 6.2). What is, however, interesting is that even Gold headlines are frequently judged as not correct, implying that either the participants were very strict, or correctness is not a necessary or particularly typical feature of newspaper headlines. While it is important for us to assess how well the generators perform also in terms of well-formed sequences, if (grammatical) correctness is not strictly a property of newspaper headlines, this evaluation question might have to be formulated differently. In any case, among the models, for the current question, the PN behaves almost on par with the gold headlines.

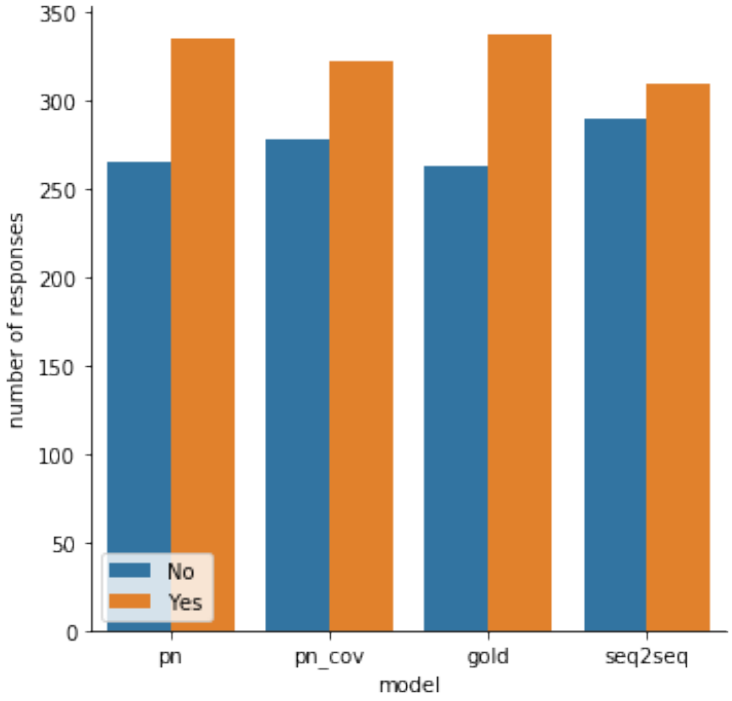


FIGURE 6.2: Correctness judgments (Q1)

Attractiveness (Q3) In the large majority of the cases, the gold headline was chosen as the most inspiring for reading the whole article (Figure 6.3). Among the models, the headlines generated by the PN is mostly chosen, followed by the PNC, and lastly by the S2S. Such results suggest that there is something in the way experts create headlines, most likely related to human creativity, rhetoric and communication strategies, which systems are not yet able to reproduce. Additionally, some online newspapers' business models can be heavily clickbait-based, causing headlines to be more sensational than faithful to the article's actual contents.

Suitability (Q4-Q5) There are two results to be analysed in the context of assessing how appropriate a headline is with respect to its article. In terms of a binary evaluation for each headline (Figure 6.4, left), in all cases, including gold, the headline is deemed not appropriate more than the times is deemed appropriate. In the case of gold, this could be due to the fact

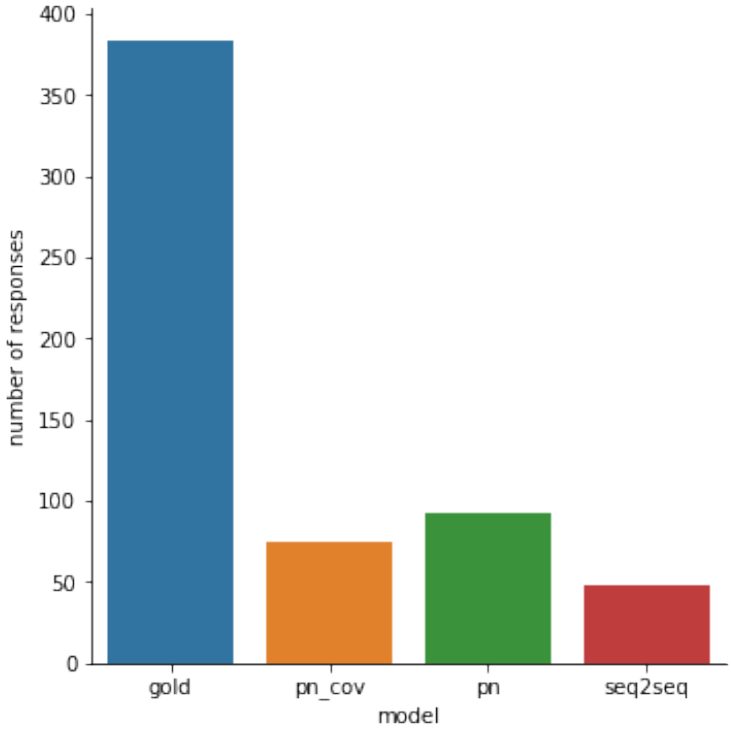


FIGURE 6.3: Attractiveness judgements (Q3)

that excessive creativity to make the title attractive can make it less adherent to the actual content. In the case of the generated headlines, they might just not be good enough.

The rank shows a possibly unexpected trend (Figure 6.4, right side). The headline chosen as most appropriate (ranked 1st) is most of the times the one produced by the PN model, even more so than the gold. Not only, but the gold is also the headline that features last (ranked 4th, thus least suitable) more than any of the other titles. This is reflected in the average rank (see caption of Figure 6.4), as the gold headline comes in last, and the PN-generated title is comparatively the most preferred.

Agreement

Given that we obtained three separate judgments per case, in addition to the separate evaluations, we can also assess how much the subjects agree with one another. Table 6.2 shows the values for Krippendorf's alpha over all of the annotated aspects. Low scores suggest that the task is highly subjective, and this is especially true for the evaluation of how attractive a headline is towards reading the whole article. Possibly surprising is the score regarding the evaluation of the headline's correctness, which could be viewed as a more objective feature to assess. Such a relatively low score could be due to the vagueness of Q1, in combination with the nature of headlines, which even in their human version might be formulated in ways that do not necessarily abide by grammatical rules.

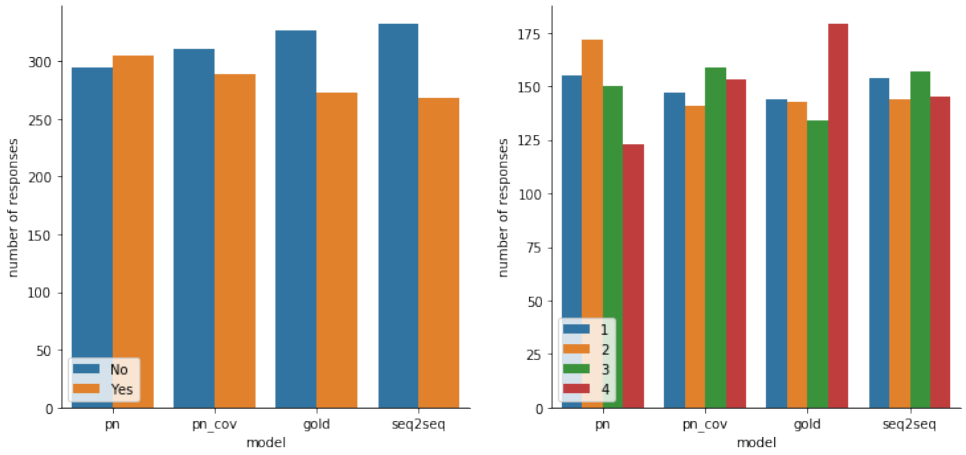


FIGURE 6.4: Suitability. Left: suitability judgment for each headline (yes/no). Right: headlines are ranked according to most (1) to least (4) appropriate for each corresponding article. Average ranking: PN=2.401; Seq2Seq=2.488; PN_C=2.530; GOLD=2.580

	G	S2S	PN	PNC	tot
correctness	0.439	0.427	0.345	0.337	0.387
attractiveness	–	–	–	–	0.120
suitability	0.349	0.354	0.374	0.313	0.348
suitability-rank	0.444	0.364	0.339	0.398	0.389

TABLE 6.2: Krippendorf's alpha scores for the human annotations. The rightmost column shows the agreement over all systems plus gold headlines. (For the attractiveness only the overall score is reported since the question asked to the annotators is "pick one")

6.2.4 Conclusions

The quality of three different sequence-to-sequence models that generate headlines starting from an article was comparatively assessed through human judgement, which we contextually used to evaluate the original headlines as well. The best system is a pointer network model, with correctness judgements on par with the gold headlines. Evaluating the generated output on different levels, especially attractiveness, which typically characterises news headlines, uncovered an interesting aspect: gold headlines appear to be the most attractive to read the whole article, but are not considered the most suitable, on the contrary, they are judged as the most unsuitable of all. Therefore, when automatically generating headlines, just relying on content might never lead us to titles that are human-like and attractive enough for people to read the article. This should be considered in any future work on news headline generation. At the evaluation stage, it would also be beneficial to involve professional journalists. The first contact with one of the newspapers at the early stages of our evaluation experiments did not yet yield any concrete collaboration, but expert judgement on the quality of the generated headlines is something we would like to include in the future.

One aspect that we have not explicitly considered in our experiments is that the headlines come from different newspapers (positioned at opposite ends of the political spectrum), and

can carry newspaper-specific characteristics. Robust headline generation should consider this, too: this topic is covered in the next section.

6.3 Embeddings Shifts as Proxies for Different Word Use in Italian Newspapers

In the previous section, we exploited human evaluation to assess the capabilities of different sequence-to-sequence model in generating newspaper headlines, as training data we used articles and headlines from two newspaper positioned at opposite ends of the political spectrum, namely *La Repubblica*[6] e *Il Giornale*[7]. We did not explicitly consider in our experiments this aspect, however, different newspapers, especially if positioned at opposite ends of the political spectrum, can render the same event in different ways. In Example (1), both headlines are about the leader of the Italian political movement “Cinque Stelle" splitting up with his girlfriend, but the Italian left-oriented newspaper *la Repubblica* (`rep` in the examples) and right-oriented *Il Giornale* (`gio` in the examples) describe the news quite differently. The news in Example (2), which is about a babysitter killing a child in Moscow, is also reported by the two newspapers mentioning and stressing different aspects of the same event.

(1) `rep` La ex di Di Maio: “E’ stato un amore intenso ma non abbiamo retto allo stress della politica"
[*en: The ex of Di Maio: “It’s been an intense love relationship, but we haven’t survived the stress of politics"*]

`gio` Luigino single, è finita la Melodia
[*en: Luigino single, the Melody is over*]

(2) `rep` Mosca, “la baby sitter omicida non ha agito da sola"
[*en: Moscow, “the killer baby-sitter has not acted alone"*]

`gio` Mosca, la donna killer: “Ho decapitato la bimba perché me l’ha ordinato Allah"
[*en: Moscow, the killer woman: “I have beheaded the child because Allah has ordered me to do it"*]

Often though, the same words are used, but with distinct nuances, or in combination with other, different words, as in Examples (3)–(4):

(3) `rep` Usa: agente uccide un nero disarmato e immobilizzato
[*en: Usa: policeman kills an unarmed and immobilised black guy*]

`gio` Oklahoma, poliziotto uccide un nero disarmato: “Ho sbagliato pistola"
[*en: Oklahoma: policeman kills an unarmed black guy: “I used the wrong gun"*]

(4) `rep` Corte Sudan annulla condanna, Meriam torna libera

[6] https://www.repubblica.it
[7] http://www.ilgiornale.it

FIGURE 6.5: Left: top 100 most frequent words in *la Repubblica*. Right: top 100 in *Il Giornale*.The words are scaled proportionally to their frequency in the respective datasets.

[*en: Sudan Court cancels the sentence, Meriam is free again*]

`gio` Sudan, Meriam è libera: non sarà impiccata perché cristiana
[*en: Sudan: Meriam is free: she won't be hanged because Christian*]

In order to have a clear understanding of these phenomena in [39] we introduced a method to study how the same words are used differently in two newspaper positioned at opposite ends of the political spectrum: *La Repubblica* and *Il Giornale*, exploiting vector shifts in embedding spaces.

The two embeddings models built on data coming from *la Repubblica* and *Il Giornale* might contain interesting differences, but since they are separate spaces they are not directly comparable. Previous work has encountered this issue from a diachronic perspective: when studying meaning shift in time, embeddings built on data from different periods would encode different usages, but they need to be comparable. Instead of constructing separate spaces and then aligning them [127], we adopt the method used by [153] and subsequently by [80] for Italian, whereby embeddings are first trained on a corpus, and then updated with a new one; observing the shifts certain words undergo through the update is a rather successful method to proxy meaning change.

Rather than across time, we update embeddings across sources that are identical in the genre (newspapers) but different in political positioning. Specifically, we train embeddings on articles coming from the newspaper *La Repubblica* (leaning left) and update them using articles coming from the newspaper *Il Giornale* (leaning right). We take the observed shift of a given word (or the shift in distance between two words) as a proxy for a difference in usage of that term, running two types of analysis. One is top-down and focuses on a set of specific words which are frequent in both corpora. The other one is bottom-up, focusing on words that result potentially interesting on the basis of measures that combine the observed shift with both relative and absolute frequency. As a byproduct, we also learn something about the interaction of shifts and frequency.

6.3.1 Data

In the scope of this work, we used the dataset already introduced in 6.2.1. For training the two word embeddings, though, we only used a selection of the data. Since we are interested in studying how the usage of the same words changes across the two newspapers, we wanted to maximise the chance of articles from the two newspapers being on the same topic. Thus, we implemented an automatic alignment and retained only the aligned news for each of the two corpora. All embeddings are trained on such aligned news.

Alignment

We align the two datasets using the whole body of the articles. We compute the tf-idf vectors for all the articles of both newspapers and create subsets of relevant news filtering by date, i.e. considering only news that was published in the range of three days before and after of one another. Once this subset is extracted, we compute cosine similarities for all news in one corpus and in the other corpus using the tf-idf vectors, we rank them and then filter out alignments whose cosine similarity is under a certain threshold. The threshold should be chosen to take into consideration a trade-off between keeping a sufficient number of documents and quality of alignment. In this case, we are relatively happy with a good but not too strict alignment, and after a few tests and manual checks, we found that the threshold of 0.185 works well in practice for these datasets, yielding a good balance between correct alignments and news recall. Table 6.3 shows the size of the aligned corpus in terms of number of documents and tokens.

newspaper	**#documents**	**#tokens**
la Repubblica	31,209	23,038,718
Il Giornale	38,984	18,584,121

TABLE 6.3: Size of the aligned corpus.

Shared lexicon

If we look at the most frequent content words in the datasets (Figure 6.5), we see that they are indeed very similar, most likely due to the datasets being aligned based on the lexical overlap.

This selection of frequent words already constitutes a set of interesting tokens to study for their potential usage shift across the two newspapers. In addition, through the updating procedure that we describe in the next section, we will be able to identify which words appear to undergo the heaviest shifts from the original to the updated space, possibly indicating a substantial difference of use across the two newspapers.

Distinguishability

Seeing that frequent words are shared across the two datasets, we want to ensure that the two datasets are still different enough to make the embeddings update meaningful.

We, therefore, run a simple classification experiment to assess how distinguishable the two sources are based on lexical features. Using the scikit-learn implementation with default parameters [233], we trained a binary linear SVM to predict whether a given document comes from *la Repubblica* or *Il Giornale*. We used ten-fold cross-validation over the aligned dataset with only word n-grams 1-2 as features and obtained an overall accuracy of 0.796, and 0.794 and 0.797 average precision and recall, respectively. This is indicative that the two newspapers can be distinguished even when writing about the same topics. Looking at predictive features we can indeed see some words that might be characterising each of the newspapers due to their higher tf-idf weight, thus maintaining distinctive context even in similar topics and with frequent shared words.

6.3.2 Embeddings and Measures

We train embeddings on one source and update the weights training on the other source. Specifically, using the `gensim` library [248], first we train a word2vec model [211] to learn 128 sized vectors on *la Repubblica* corpus (using the skip-gram model, a window size of 5, high-frequency word downsample rate of 1e-4, a learning rate of 0.05 and minimum word frequency 3, for 15 iterations). We call these word embeddings $spaceR$. Next, we update $spaceR$ on the documents of *Il Giornale* with identical settings but for 5 iterations rather than 15. The resulting space, $spaceRG$, has a total vocabulary size of 53,684 words. We decided to go this direction (rather than train on Il Giornale first and update on La Repubblica later because the La Repubblica corpus is larger in terms of tokens, thus ensuring a more stable space to start from.

Quantifying the shift

This procedure makes it possible to observe the shift of any given word, both quantitatively as well as qualitatively. This is more powerful than building two separate spaces and just check the nearest neighbours of a selection of words. In the same way that the distance between two words is approximated by the cosine distance of their vectors [295], we calculate the distance between a word in $spaceR$ and the same word in $spaceRG$, by taking the norm of the difference between the vectors. This value for word w is referred to as $shift_w$. The higher $shift_w$, the larger the difference in usage of w across the two spaces. We observe an average shift of 1.98, with the highest value at 6.65.

Frequency impact

By looking at raw shifts, selecting high ones, we could see some potentially interesting words. However, frequency plays an important role, too [269]. To account for this, we explore the impact of both absolute and relative frequency for each word w. We take the overall frequency of a word summing the individual occurrences of w in the two corpora ($total_w$). We also make the difference between the relative frequency of a word in the two corpora, as this

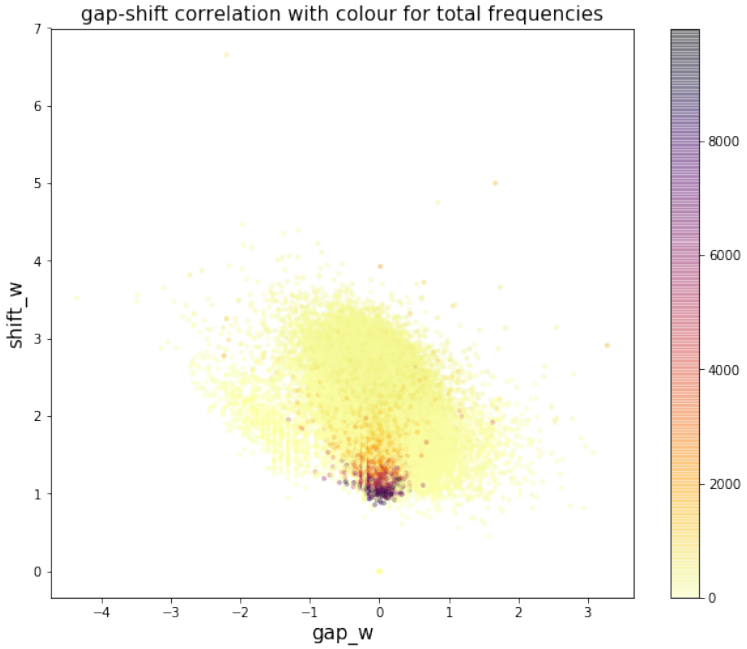


FIGURE 6.6: Gap-Shift scatters plot of the words in the two newspapers. Darker colour indicates a higher cumulative frequency; a negative gap means higher relative frequency in *Il Giornale*.

might be influencing the shift. We refer to this difference as gap_w, and calculate it as in Equation 6.1.

$$gap_w = log(\frac{freq^r_w}{|r|}) - log(\frac{freq^g_w}{|g|}) \tag{6.1}$$

A negative gap_w indicates that the word is relatively more frequent in *Il Giornale* than in *la Repubblica*, while a positive value indicates the opposite. Words whose relative frequency is similar in both corpora exhibit values around 0.

We observe a tiny but significant negative correlation between $total_w$ and $shift_w$ (-0.093, $p < 0.0001$), indicating that the more frequent a word, the less it is likely to shift. In Figure 6.6 we see all the dark dots (most frequent words) concentrated at the bottom of the scatter plot (lower shifts).

However, when we consider gap_w and $shift_w$, we see a more substantial negative correlation (-0.306, $p < 0.0001$), suggesting that the gap has an influence on the shift: the more negative the gap, the higher the shift. In other words, the shift is larger if a word is relatively more frequent in the corpus used to update the embeddings.

6.3.3 Analysis

We use the information that derives from having the original $spaceR$ and the updated $spaceRG$ to carry out two types of analysis. The first one is top-down, with a pre-selection of words to study, while the second one is bottom-up, based on measures combining the shift and frequency.

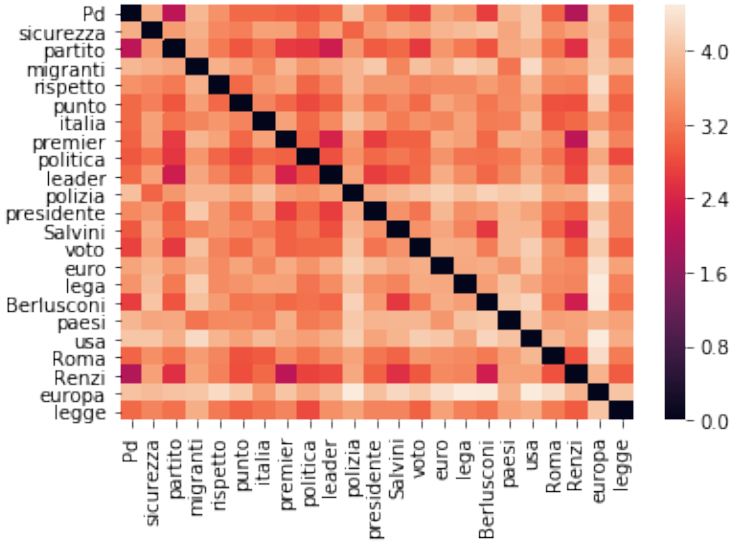


FIGURE 6.7: Distance matrix between a small set of high frequency words on *la Repubblica*. The lighter the colour the larger the distance.

Top-down

As a first analysis, we look into the most frequent words in both newspapers and study how their relationships change when we move from $spaceR$ to $spaceRG$. The words we analyse are the union of those reported in Figure 6.5. Note that in this analysis we look at pairs of words at once, rather than at the shift of a single word from one space to the next. We build three matrices to visualise the distance between these words.

The first matrix (Figure 6.7) only considers $SpaceR$, and serves to show how close/distant the words are from one another in *la Repubblica*. For example, we see that “partito" and “Pd", or “premier" and “Renzi" are close (dark-painted), while “polizia" and “europa" are lighter, thus more distant (probably used in different contexts).

In Figure 6.8 we show a replica of the first matrix, but now on $SpaceRG$; this matrix now lets us see how the distance between pairs of words has changed after updating the weights. Some vectors are farther than before and this is visible by the lighter colour of the figure, like “usa" and “lega" or “italia" and “usa", while some words are closer like “Berlusconi" and “europa" or “europa" and “politica" which feature darker colour. Specific analysis of the co-occurrences of such words could yield interesting observations on their use in the two newspapers.

In order to better observe the actual difference, the third matrix shows the shift from $spaceR$ to $spaceRG$, normalised by the logarithm of the absolute difference between the $total_{w1}$ and $total_{w2}$ (Figure 6.9).[8] Lighter word-pairs shifted more, thus suggesting different contexts and usage, for example “italia" and “lega". Darker pairs, on the other hand, such as “Pd"-“Partito" are also interesting for deeper analysis, since their joint usage is likely to be quite similar in both newspapers.

[8]Note that this does not correspond exactly to the gap measure in Eq. 6.1 since we are considering the difference between two words rather than the difference in the occurrence of the same word in the two corpora.

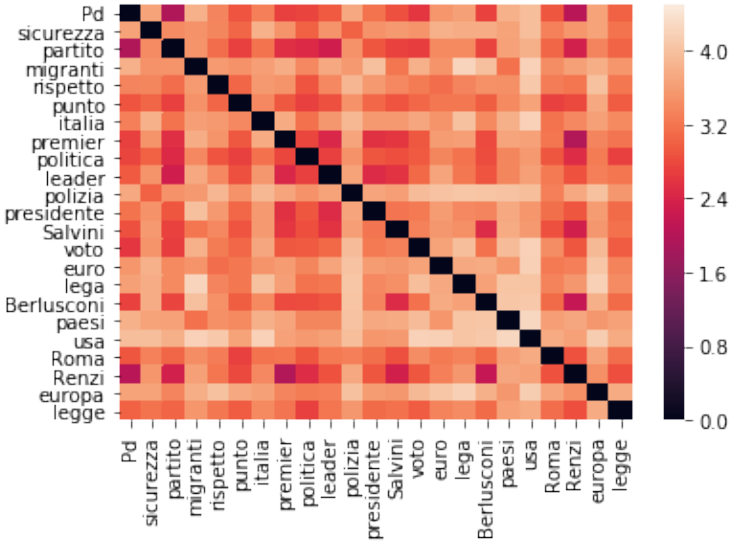


FIGURE 6.8: Distance matrix between a small set of high frequency words after updating with *Il Giornale*. The lighter the colour the larger the distance.

Bottom-up

Differently from what we did in the top-down analysis, here we do not look at how the relationship between pairs of pre-selected words changes, rather at how a single word's usage varies across the two spaces. These words arise from the interaction of gap and $shift$, which yields various scenarios. Words with a large negative gap (relative frequency higher in *Il Giornale*) are likely to shift more, but it's probably more of an effect due to increased frequency than a genuine shift. Words that have a high gap (occurring relatively less in *Il Giornale*) are likely to shift less, most likely since adding a few contexts might not cause much shift.

The most interesting cases are words whose relative frequency does not change in the two datasets but have a high shift. Zooming in on the words that have small gaps ($-0.1 < gap_w < 0.1$), will provide us with a set of potentially interesting words, especially if they have a shift higher than the average shift. We also require that words obeying the previous constraints occur more than the average word frequency over the two corpora. Low-frequency words are in general less stable [269], suggesting that shifts for the latter might not be reliable. High-frequency words shift globally less (cf. Figure 6.6), so a higher than average shift could be meaningful.

Figure 6.10 shows the plot of words that have more or less the same relative frequency in the two newspapers ($-0.1 < gap > 0.1$ and an absolute cumulative frequency higher than average), and we, therefore, infer that their higher than average shift is mainly due to usage difference. Some comments are provided next to the plot.

These words can be the focus of a dedicated study, and independently of the specific observations that we can make in this context, this method can serve as a way to highlight the hotspot words that deserve attention in a meaning shift study.

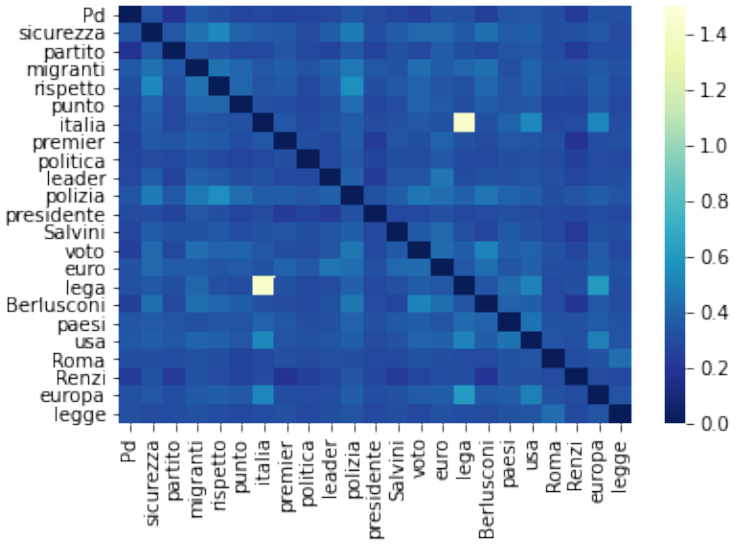


FIGURE 6.9: Difference matrix between embeddings from *spaceR* and *spaceRG* normalised with the logarithm of the absolute frequency difference in *spaceRG*. The lighter the colour, the larger the distance between pairs of words.

A closer look at nearest neighbours

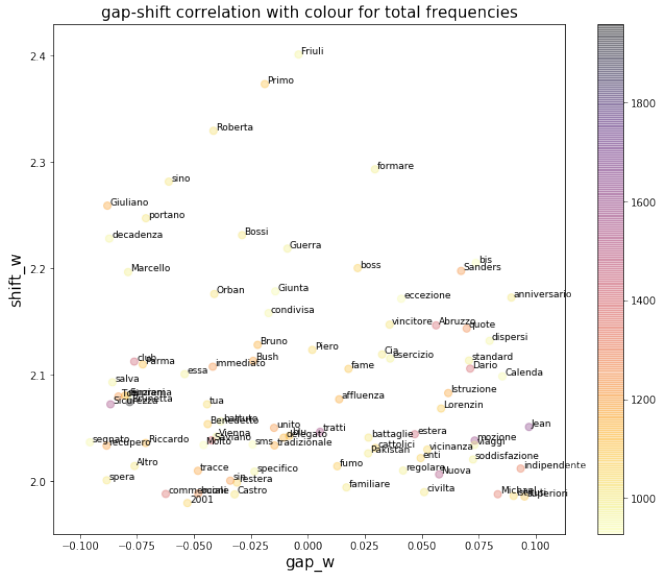
As a last, more qualitative, analysis, one can inspect how the nearest neighbours of a given word of interest change from one space to the next. In our specific case, we picked a few words (deriving them from the top-down, thus most frequent, and bottom-up selections), and report in Table 6.4 their top five nearest neighbours in *SpaceR* and in *SpaceRG*. As in most analyses of this kind, one has to rely quite a bit on background and general knowledge to interpret the changes. If we look at “Renzi", for example, a past Prime Minister from the party close to the newspaper “la Repubblica", we see that while in $SpaceR$ the top neighbours are all members of his own party, and the party itself (“Pd"), in $SpaceRG$ politicians from other parties (closer to “Il Giornale") get closer to Renzi, such as Berlusconi and Alfano.

6.3.4 Conclusions

We experimented with using embeddings shifts as a tool to study how words are used in two different Italian newspapers. We focused on a pre-selection of high-frequency words shared by the two newspapers, and on another set of words that were highlighted as potentially interesting through a newly proposed methodology that combines observed embeddings shifts and relative and absolute frequency. The most differently used words in the two newspapers are proper nouns of politically active individuals as well as places, and concepts that are highly debated on the political scene.

Besides the present showcase, we believe this methodology can be more in general used to highlight which words might deserve deeper, dedicated analysis when studying meaning change.

One aspect that should be further investigated is the role played by the methodology used for aligning and/or updating the embeddings. As an alternative to what we proposed, one could



Out of the resulting 89 words, 28 are named entities, including politician (Bossi, Lorenzin, Castro, Bush, Saviano, Orban), organisation (CIA, Istruzione[a], Nuova[b]), and location names (Friuli, Vienna, Parma, Pakistan). Common nouns are mostly related to political irksome aspects on which the two newspapers might indeed take diverging positions, such as "sicurezza", "estera", "boss". Other words clearly related to political positions are "renziani", "cattolici", and "tradizionale", the last two being probably part of the 2019 political debate on the so-called traditional family in Italy.

[a]Most likely a token from the expression "Ministero dell'Istruzione, dell'Università e della Ricerca.

[b]Most likely a token of the bigram "Forza Nuova", an extreme right political movemement.

FIGURE 6.10: Gap-Shift scatter plot like in Figure 6.6, zoomed in the gap region -0.1 - 0.1 and shift greater than 1.978 (average shift). Only words with cumulative frequency higher than average frequency are plotted.

TABLE 6.4: A few significant words and their top 5 nearest neighbours in *SpaceR* and *SpaceRG*.

SpaceR	*SpaceRG*
"migranti" [*en: migrants*]	
barconi [*large boats*] (0.60)	eritrei [*Eritreans*] (0.61)
naufraghi [*castaways*] (0.57)	Lampedusa [] (0.60)
disperati [*wretches*] (0.56)	accoglienza [*hospitality*] (0.59)
barcone [*large boat*] (0.55)	Pozzallo [] (0.58)
carrette [*wrecks*] (0.53)	extracomunitari [*non-European*] (0.57)
"Renzi " [past Prime Minister]	
Orfini [] (0.65)	premier [] (0.60)
Letta [] (0.64)	Nazareno [] (0.59)
Cuperlo [] (0.63)	Berlusconi [] (0.58)
Pd [] (0.62)	Cav [] (0.57)
Bersani [] (0.61)	Alfano [] (0.56)
"politica " [*en: politics*]	
leadership [] (0.65)	tecnocrazia [*technocracy*] (0.60)
logica [*logic*] (0.64)	democrazia [*democracy*] (0.59)
miri [*aspire to*] (0.63)	partitica [*of party*] (0.58)
ambizione [*ambition*] (0.62)	democratica [*democratic*] (0.57)
potentati [*potentates*] (0.61)	legalità [*legality*] (0.56)

employ different strategies to manipulate embedding spaces towards highlighting meaning changes. For example, [261] exploited Representational Similarity Analysis [163] to compare

embeddings built on different spaces in the context of studying diachronic semantic shifts in ancient Greek. Another interesting approach, still in the context of diachronic meaning change, but applicable to our datasets, was introduced by [126], who use both a global and a local neighbourhood measure of semantic change to disentangle shifts due to cultural changes from purely linguistic ones.

6.4 Invisible to People but not to Machines: Evaluation of Style-aware Headline Generation in Absence of Reliable Human Judgment

Automatic headline generation is conceptually a simple task that can be conceived as a form of extreme summarisation [265]: given an article or a portion of it, generate its headline. As we have seen in previous sections the task can therefore be seen as extreme summarisation. The generation of headlines though is not just a matter of summarizing the content. As we seen in the previous section different newspapers report the news in different ways, depending on their policies and strategies. For example, they might exhibit some topic-biases, such as writing more about gossip vs more about politics. But even when reporting on the same topics, they might exhibit specific stylistic features related to word choices, word order, punctuation usage, etc. Such newspaper-specific style is likely to be exhibited not only in the articles' body but also in the headlines, which are a prime tool to capture attention and make clear statements about the newspaper's position over a certain event.

Can this newspaper-specific style be distinguished? And is it preserved in automatically generated headlines? To answer such questions, in [79] we train newspaper-specific headline generation models and evaluate how style-compliant the generated headline is for a given newspaper. How such evaluation can be performed though is yet another research question of its own.

Evaluating generated text just using standard metrics based on lexical overlap is normally not accurate enough [182]. In machine translation, for example, the decisive, final system evaluation is typically human-based, as the lexically-based BLEU score is not exhaustive. Automatic evaluation strategies are still used because human evaluation is expensive, not always available, and complex to include in highly iterative developments. However, human evaluation is not always a decisive and accurate strategy, since there might be aspects of a text that for people are not so easy to grasp. For example, in profiling, where differently from the assessment of the goodness of translated text, evaluation can be performed against discrete gold labels, several studies found that humans are definitely not better than machines in identifying the gender of a writer [161, 100, 118]. Similarly, humans failed to outperform automatic systems in recognising the native language of non-English speakers writing in English [200]. [14] also found that seven out of ten subjects, including professional translators, performed worse than a simple SVM at the task of telling apart original from translated texts.

More generally, [105] have observed that it is difficult to ascertain if readers can perceive subtle *stylistic variations*, and past human-based evaluations of style have indeed shown very

low inter-rater agreement [23, 42, 85]. In spite of a recent surge of works focusing on style in generation [95, 138, 151, e.g.], and on attempts to define best practices for human and automatic evaluation [174], reliable and shared evaluation metrics and strategies concerning style-aware generation are still lacking [102].

As a contribution to this aspect, we develop style-aware headline generation models, and discuss an evaluation strategy based on text classification, which is particularly useful given that human judgement for this task is found to be unreliable. While the strategy of using classification as evaluation is in itself not new, this work has a series of innovative aspects which we discuss in the context of related work (Section 6.4.1).

Contributions. (i) we develop and share models based on a pointer network with coverage attention to generate newspaper-specific headlines for two Italian newspapers given the article; (ii) we show that an automatic, classification-based methodology can be used to evaluate style-compliance in NLG, and can successfully substitute human judgement which proves to be unreliable for this task.

6.4.1 Related Work

The focus of this contribution is not on investigating the best models for style-compliant headline generation. Rather, we want to test an automatic evaluation strategy that can overcome the limitation of unreliable human judgement. Besides the works mentioned in the Introduction to frame the problem, we will not discuss further related work on style modelling or summarisation. Rather, we concentrate on discussing previous works that make use of automatic classification for the evaluation of NLG systems, also to show in what sense our approach differs from existing ones.

Using a classifier to assess the goodness of generated texts in connection to a broad definition of style-aware generation has been used in several previous works [138, 292, 241, 145, 178, e.g.]. However, these works tend to focus on sentiment aspects (transforming a positive review into a negative one, for example), which are usually mostly associated to a lexical problem (only a small part of *style*). Indeed, the problem of style transfer is usually addressed within the Variational Autoencoder framework and/or through lexical substitution. The lexical substitution was also the key element of a system developed for obfuscating gender-related stylistics aspects in social media texts [247], where a classification-based evaluation was used.

In addition, [178] compared the automatic classification-based evaluation with human evaluation. They find a high correlation between human and automatic evaluation in two out of their three data-sets, showing the validity of the automatic approach. However, the task of sentiment analysis, though subjective, is not too hard for humans, who are usually able to perceive sentiment encapsulated in text. [246] also exploited human and automatic classification as benchmarks for a machine translation system that translates formal texts into informal texts and vice-versa. Also in this case, usually text register is something that humans are quite able to grasp.

Our work differs from the above in at least two aspects. One is that we want to evaluate the capabilities of an NLG system to learn (different) stylistics aspects from (different) training data sets, rather than evaluating the capabilities of style transfer systems mostly based on lexical substitution. The other is that the stylistic aspects that we attempt to model are not easily identified by human annotators. Therefore, relying on human-based evaluation in a real setting is not an option, and even the classification-based method cannot be easily validated against human judgement for this task. Also because of this, we devised a quite fine-grained evaluation setting, carefully selecting training and testing conditions.

6.4.2 Approach and Models

The principle behind our approach is using a classifier to assess the style-compliance of automatically generated text.

Specifically, we train two models to generate headlines for newspaper articles coming from two (politically) different newspapers, namely *La Repubblica* (left-wing), and *Il Giornale* (right-wing), and expect that the generated headlines will carry some newspaper-specific characteristics (see also [239, 293]).

At the same time, on the gold headlines from the two newspapers, we train a prediction model that learns to classify a given headline as coming from one newspaper or the other. The good performance of this classifier indicates that it is able to distinguish the two sources.

In order to test whether the generation is indeed newspaper-specific, we run the classifier on the automatically generated headlines and verify whether it is able to correctly classify their source.

Figure 6.11 shows an overview of the approach.

Generation Models

As the focus of this contribution is not on making the best model for headline generation, rather on evaluation strategies, we leverage existing implementations of sequence-to-sequence networks. More specifically, we experiment with the following three models:

- *Sequence-to-Sequence with Attention (S2S)*
  We used a sequence-to-sequence model [288] with attention [8] to the configuration used by [272] but we used a bidirectional instead of a unidirectional layer. This choice applies to all the models we used. The final configuration is 1 bidirectional encoder-decoder layer with 256 LSTM cells each, no dropout and shared embeddings with size 128; the model is optimised with Adagrad with learning rate 0.15 and gradient clipped [209] to a maximum magnitude of 2.

- *Pointer Generator Network (PN)*
  The basic architecture is a sequence-to-sequence model, but the hybrid pointer-generator network uses a *pointing mechanism* [272] that lets it copy words from the source text,

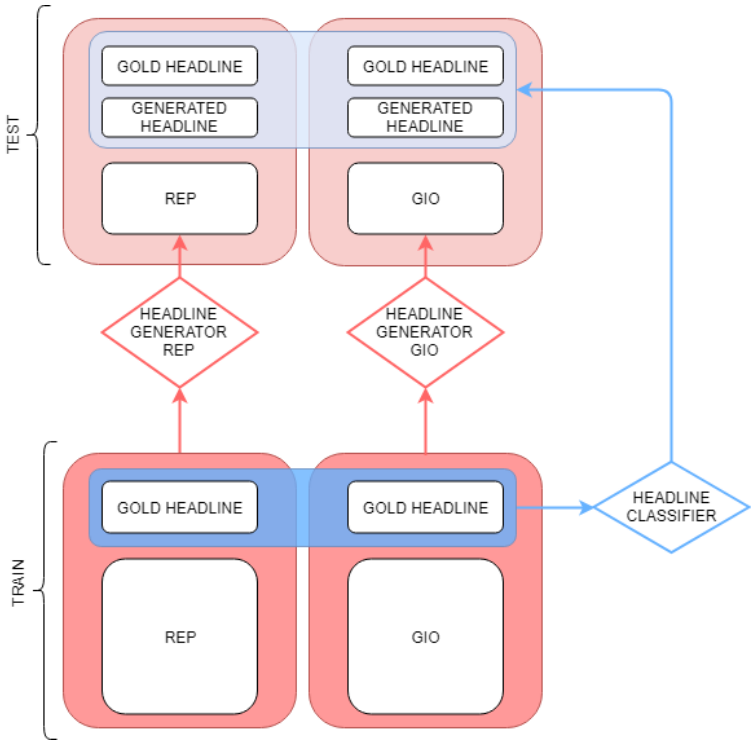


FIGURE 6.11: Red: generation task. Blue: classification task. Darker: training. Lighter: testing.

and generate words from a fixed vocabulary. This allows for better handling of out-of-vocabulary words, providing accurate reproduction information while retaining the ability to reproduce novel words.

- *Pointer Generator Network with Coverage (PNC)*
  This model is basically a Pointer Generator Network with an additional coverage attention mechanism that is intended to overcome the copying problem typical of sequence-to-sequence models. This is done by penalising the attention over already generated words [272].

In order to assess the quality of the generated headlines, independently of whether they were maintaining or not the style of the source, we ran a human-based evaluation on a variety of criteria, including grammatical correctness and appropriateness to the article's content (for details see [39]).

Results showed that while the basic sequence-to-sequence model produces rather low-quality headlines, the pointer network, with and without attention, yields headlines whose grammaticality is on par with the gold, human-written headlines.[9] Automatically generated headlines apparently are not as attractive towards reading the whole paper as the gold headlines but compared to the latter they were evaluated much more appropriate in terms of reflecting the article's content.

[9]Please note that in any case, humans do not judge either gold or automatically produced headlines as particularly correct according to grammatical standards, as grammatical correctness per se is not necessarily a requirement of news' titles [40].

	Rep F1	**Gio F1**	**AVG F1**
classifier	0.813	0.812	0.813
human	0.619	0.640	0.630

TABLE 6.5: Classification performance on random split.

For the current evaluation experiments, we thus opt for a pointer network with coverage attention and generate headlines according to different newspapers' styles. We train two pointer network models that, given the first portion of an article (approx. 500 words), learn to generate its respective headline. The first model is trained on articles from *la Repubblica*, while the second model is trained on *Il Giornale*. From an architecture and implementation perspective, the models and their parameters are identical.

Classifier

We use a Bidirectional LSTM (Bi-LSTM) [135] which exploits as features the concatenation of word and character embeddings. We used a word embeddings lexicon trained with word2vec [211] on the ItWac Corpus [15] in a previous work by [58]. The character embeddings are extracted by a Convolutional Neural Network (CNN) [173] that takes as input a sequence of one-hot encoded characters. The CNN weights are optimised during training. We use a sigmoid layer as a classifier.

For each training setting (see Section 6.4.4), we extracted a randomly sampled validation set (10% of the training set) which we used for model selection and fine-tuning. We use binary cross-entropy as a loss function, and the Adam optimiser [154] for optimisation.

6.4.3 Data

We exploited again the dataset introduced in 6.2.1.

For our experiments, we want to account for potential topic biases in the two newspapers and reduce them as much as possible. This should help us to better disentangle newspaper-specific style from potential newspaper-specific topics. Thus, we create a subset of the data where articles are topic-aligned.

Alignment

While we work with headlines, the alignment procedure is run over the whole articles. This is exactly because we want the headlines to refer to the same topics, but we know that they might not express the same content in the same way. Thus, we expect that headlines of aligned articles might not necessarily be that similar (see indeed also examples in Table 6.6).

First, we clean the full articles, removing stop words and punctuation. Second, we compute the tf-idf vectors of all the articles of both newspapers and we create subsets of relevant news filtering by date, i.e. considering only news which was published in approximately the same, short, temporal range for the two sources. Third, on the tf-idf vectors we compute cosine

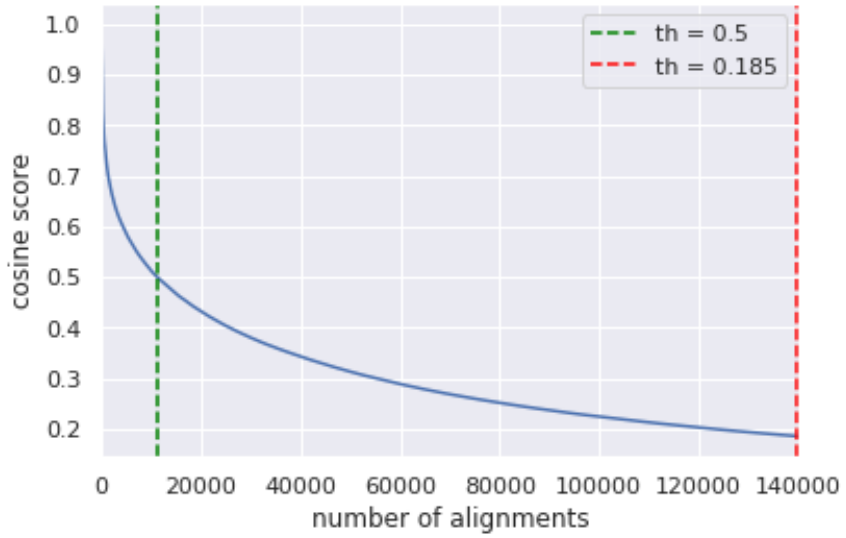


FIGURE 6.12: Trend of the number of alignments varying with the cosine similarity threshold. The green vertical dashed line is the stricter threshold, used to get the best alignments, the red one is the looser one.

similarities for all news in the resulting subset. Fourth, we rank them and retain only the alignments that are above a certain threshold.

The threshold is chosen to take into consideration a trade-off between the number of documents and the quality of alignments. The quality is assessed by manual inspection of random samples. In this experiment we choose two different thresholds: one is stricter (> 0.5) and we use it to select best alignments for the test set; the other one is looser (> 0.185, and $<= 0.5$) and we use it to select a portion of alignments to use in one of the training sets we experiment with (train-M, see Section 6.4.3 below).

In Figure 6.12 we show the trade-off between the strictness (in terms of cosine similarity) and the number of alignments. As can be expected, the number of alignments exponentially grows when decreasing the similarity score. Our stricter threshold (the green dashed line, 0.5) guarantees high-quality alignments, while the looser one (the red dashed line, 0.185) provides a large number of at least partially aligned news. As a quality control, we observe that restricting the considered news to a short time span makes it possible to obtain reliable alignments even with a relatively low similarity threshold while preserving some substantial number of instances, which we need to use for training. In Table 6.6 we report some examples of aligned headlines with varying similarity scores. As mentioned before, while articles might exhibit a high lexical overlap which has indeed led to strict alignment (> 0.5), the *La Repubblica*'s headline might be very different from the one written by *Il Giornale*, highlighting different aspects of the news in different ways.

Test set

The test set stays the same across all settings.

It contains only aligned headlines (11k total), which are selected after the alignment procedure described in Section 6.4.3 as having a minimum cosine distance of 0.5, thus

cosine score	newspaper	alignment
0.96	rep	Estroverso o nevrotico? Lo dice la foto scelta per il profilo social *en:[Extrovert or neurotic? The photo chosen for the social profile says so]*
	gio	L'immagine del profilo usata nei social network rivela la nostra personalità *en:[The profile picture used in social networks reveals our personality]*
0.5 (strict)	rep	Egitto, governo si dimette a sorpresa *en:[Egypt, government resigns surprisingly]*
	gio	Egitto, il governo si dimette *en:[Egypt, government resigns]*
0.185 (loose)	rep	Elezioni presidenziali Francia, la Chiesa non si schiera né per Macron né per Le Pen *en:[Presidential elections France, the Church does not take sides either for Macron or for Le Pen]*
	gio	Il primo voto con l'incubo Isis ma il terrorismo esce sconfitto *en:[The first vote with the Isis nightmare but terrorism comes out defeated]*

TABLE 6.6: Example of alignments between *La Repubblica* and *Il Giornale*, extracted with different similarity scores. The second and the third ones are respectively the strict and the loose threshold used to split the alignments. The first two headlines are well aligned, the third one has a partial alignment.

ensuring their articles are lexically very similar. The rationale behind this is that testing on aligned data tries to remove a topic factor: if the classifier is able to distinguish generated headlines from the two newspapers in spite of them coming from the lexically aligned dataset, these headlines are likely to carry some characteristics of the two newspapers that are not necessarily topic-related.

Training sets

We create two different training sets of equal size, each composed of a total of 130K documents: 65K from *la Repubblica* and 65K from *Il Giornale*. These two training sets differ with respect to alignment and therefore potential topic bias:

- **train-D**, where we exclude all aligned data, resulting in a topic biased dataset since the two newspapers often focus on different topics (*Il Giornale* for example has much more gossip than *la Repubblica*);
- **train-M**, where we include weakly aligned data (cosine distance between 0.185 and 0.5), resulting in a mixed, less topic biased dataset; train-M is, therefore, more similar than train-D to the test set (which, as explained, only includes strongly aligned texts).

Please note that each training set contains two equally represented portions of the two newspapers. Thus train-D contains a subset of *la Repubblica* and a subset of *Il Giornale*, and likewise for train-M.

6.4.4 Classification as Evaluation

Given that we want to train models that are able to generate headlines retaining the specific style of a given newspaper, we will know that we are successful if indeed our automatically generated headlines can be recognised as pertaining to one and not the other source.

In this Section we outline our approach to performing this non-trivial evaluation and the results we obtain.

	in-generator	**cross-generator**
train-D	setting 1	setting 3
train-M	setting 2	setting 4
test set = same and aligned for all settings		

TABLE 6.7: Experimental settings. In **train-D** all of the aligned data is excluded; in **train-M** the data is mixed, thus also including weakly aligned texts (highly aligned data is only used in the test set). The two trainsets are equal in size, and the two corpora therein are balanced, too. In **cross** settings we use the model trained on one newspaper to generate headlines from articles of the other newspaper.

Automatic vs Human Classification

A first option is to ask humans to perform this evaluation, but as mentioned, humans have proven not much reliable in capturing stylistic aspects [23, 42, 85, 105]. A second option is to do this evaluation automatically, but we need to have reliable models that are able to distinguish the two sources/styles.

In order to assess the classifier's ability to correctly label the headlines from the two newspapers, we randomly split our gold data into 80% training and 20% test (no generated data is involved at this stage, and no information about news alignment is exploited). As a preliminary test, we asked one annotator (largely familiar with one of the two newspapers) to label 100 gold headlines randomly picked to get a first idea of the task's feasibility.

Results for both model and the human judge are reported in Table 6.5. We take them as a general indication that (i) headlines are indeed classifiable automatically with good accuracy, (ii) humans seem not as reliable at the same task.

At this stage though we do not know if the classifier's ability is related to detecting the newspapers' specific styles or rather content. Indeed, the classification model is trained on non-aligned data, and thus potentially topic biased. We, therefore, design our experiments using different training strategies and splits, but a single test set across all settings, in order to best evaluate newspaper-specific style, rather than content. We also include more humans in the evaluation loop, for comparison and to further verify their ability at this task.

Settings

We generate and classify headlines under the four different settings shown in the matrix in Table 6.7.

Training Generation Models For generation, in all settings we always train two distinct generation models: one on the *la Repubblica* data, which learns to generate *la Repubblica*-specific headlines, and one on the *Il Giornale* portion of the documents, learning to generate *Il Giornale*-specific headlines. In setting1-3 the training is done over the topic-biased training sets (train-D), and in setting2-4 over the mixed datasets (train-M).

Applying Generation Models on the Testset When generating headlines, we use two conditions, according to whether the generation model is tested on articles from the same newspaper it was trained on (*in-newspaper*, settings1-2) or not (*cross-newspaper*, settings3-4).

In settings1-2, we use each generator on its own test set: we ran the *la Repubblica* model over *la Repubblica* articles in the test set and generated the corresponding headline. Likewise for *Il Giornale*.

In settings3-4, instead, we cross-test the models: we run the *la Repubblica* model over *Il Giornale* articles in the test set, and generate the corresponding headline. Even though the articles come from the other newspaper, we expect that the model, if it has learnt appropriately, still tries to come up with a *la Repubblica*-specific title. We did the same with *Il Giornale* model, running it over *la Repubblica* test set.

Evaluating Generation Models through Classification For classification, we trained two classifiers: one on the topic-biased train-D (settings1-3), the other on the mixed train-M (settings2-4). At the classification stage, we assess the performance of the generators using the respective classifier for each setting over the following headlines:

1. a validation set which comes from the same distribution of each training set;
2. gold headlines in the test set;
3. generated headlines in the test set:
    - in *settings*1-2 we test in-newspaper generated headlines;
    - in *settings*3-4 we test cross-newspaper generated headlines.

In each case, we assess the influence of topic bias and similarity between training and test set by testing both the model trained on train-D and that trained on train-M.

Expectations

The experiments were designed and run with the following expectations for the classification models:

E1 reasonable classification performance (above 50% baseline) on the generated headlines in all settings, indicating that the generators are able to capture newspaper-specific traits and reproduce them in the generated headlines. We expect in any case the performance to be lower than on gold headlines in the same setting;

E2 better classification performance on the generated headlines in setting2 than in setting1, as the test set is strict-aligned, thus topic-unbiased, while train-D (setting1) is highly topic-biased;

E3 worse classification performance on gold headlines of the test sets than those of the validation sets as the latter comes from the same distribution as the training sets, while the test set is strict-aligned; this is especially true for setting1, where we expect a larger

	Ann 1	Ann 2	Ann 3	Agreement
gold	0.58	0.62	0.57	0.16
setting 1	0.57	0.59	0.54	0.14
setting 2	0.57	0.60	0.56	0.13

TABLE 6.8: Annotators' accuracy and agreement on sampled aligned test sets. The agreement is computed as Krippendorff's Alpha Reliability.

Test set	Rep F1	Gio F1	AVG F1
train-D (settings1-3)			
validation	0.819	0.815	0.817
gold	0.755	0.703	0.729
in-generated (setting1)	0.701	0.630	0.666
cross-generated (setting3)	0.682	0.548	0.615
train-M (settings2-4)			
validation	0.810	0.809	0.810
gold	0.782	0.770	0.776
in-generated (setting2)	0.690	0.653	0.672
cross-generated (setting4)	0.646	0.567	0.607
human evaluation on sample from test set			
gold (avg)	0.543	0.620	0.582
in-generated (setting1) (avg)	0.600	0.527	0.563
in-generated (setting2) (avg)	0.607	0.530	0.569

TABLE 6.9: Results for the different experiments.

gap between validation and test; the gap should be smaller in setting2 since the training set is closer to the test set;

E4 good performance on the cross-generated headlines (settings3-4), showing that a newspaper's style is preserved in headlines even when generated from articles of a different newspaper, though lower than the classification performance of the in-newspaper generation (settings1-3). The smaller the difference between setting1 and setting3 (and setting2 and setting4), the better the model captures newspaper-specific stylistic features.

Results

We discuss the classifiers' results in relation to our expectations. Before doing so, we run a few more human-based evaluations, which we report on first.

In order to further assess the human ability to distinguish headlines from the two newspapers in the same settings of the classifiers (rather than a random split as briefly reported in Section 6.4.4 above), we asked three annotators to label 200 gold headlines each picked randomly from the aligned test set (100 from *la Repubblica*, 100 from *Il Giornale*). Also, we asked the annotators to label 200 headlines generated automatically in setting 1 and setting 2.

example	generated	newspaper	human pred	machine pred
Usa - Cuba , Obama : " Bienvenido a Cuba " . E l' Avana accoglie tre giorni *en: [Usa - Cuba, Obama: "Bienvenido a Cuba". And Havana welcomes three days]*	Yes	rep	rep	gio
La verita su Twitter : " Macchina del fango " . Ma il Pdl è insorto *en: [The truth on Twitter: "Mud Machine". The PDL has arisen]*	Yes	gio	gio	gio
De Benedetti : " Riforma Popolari , tutta la storia di Pulcinella " . Il Pd : " Ne parlavano tutti " *en: [De Benedetti: "Populars reform, the whole story of Pulcinella", PD: "Everyone was talking about it"]*	Yes	rep	gio	rep
Rai verso le nomine per le reti : ecco i nomi *en: [Rai towards the nominations for the channels: here are the names]*	No	gio	gio	rep
Nasa , la Terra ha sette " sorelle " : scoperto un nuovo sistema planetario *en: [Nasa, the Earth has six "sisters": a new planetary system is discovered]*	No	rep	rep	rep
Vaccino antinfluenzale : ecco i cinque miti da sfatare *en: [Flu vaccine: here are the five myths to dispel]*	No	gio	rep	gio

TABLE 6.10: Examples of human and automatic evaluation of gold and generated headlines. The examples are randomly picked from any setting.

model	example
gold (rep)	Erdogan - Netanyahu , accuse durissime : " Israele come Hitler " , " No , tu sei un dittatore e stragista " (Erdogan - Netanyahu , very serious accusations : " Israel like Hitler " , " No , you are a dictator and mass killer ")
rep_D	Erdogan - Israele , la replica : " Israele e il Paese piu fascista "
rep_M	Israele , Netanyahu : " Israele e il Paese piu sionista , Hitler fascista fra i curdi "
gio_D2rep	Erdogan : " Premier razzista del mondo " Il piano di accuse per i curdi
gio_M2rep	Erdogan : " Il Paese piu sionista , razzista del mondo " . La replica araba
gold (gio)	Ecco le cellule hackerate per sconfiggere il cancro (Here are the hacked cells to defeat cancer)
gio_D	Il Mit di Boston : " Hackerare e riprogrammare le cellule per combattere il cancro "
gio_M	Hackerare le cellule per il cancro ' : ' riprogrammare il Dna '
rep_D2gio	Boston , ecco il codice genetico per combattere i tumori . " E ora un linguaggio "
rep_M2gio	Il Mit di un codice del Dna : cosi possibile hackerare le cellule sane e riprogrammarle

TABLE 6.11: Examples of generated headlines in the different settings for *la Repubblica* and *Il Giornale*

This evaluation is therefore directly comparable with the automatic evaluation over the gold data and the generated headlines in the corresponding settings. All annotators are familiar with at least one of the two newspapers.[10]

The results reported in Table 6.8 show that human annotators definitely do not perform well at distinguishing the gold headlines, not much above the 50% baseline. Similar scores are observed in the assessment of the automatically generated headlines for both settings. Also, the level of agreement (computed as the Krippendorff's Alpha Reliability) is very low for both gold and generated headlines, further indicating that human evaluations are not reliable for this task. To provide a few concrete examples, in Table 6.10 we show some gold and generated headlines together with their human and automatic evaluation.

Table 6.9 reports the results for all settings and the average of the performance of the three human evaluators for comparison.

Regarding **E1**, we indeed observe that for gold headlines the performance of the classifier is higher than for generated headlines, although for all the generated headlines the classifier performance is significantly higher than a random baseline. This suggests that the generators are able to intercept stylistic features and generate text accordingly.

[10]We did seek a collaboration with expert title creators for one of the two newspapers, as they are likely to have a different perception of the headlines, but received a negative response. We discuss this further in Section 6.4.5 in the context of future work.

Also, **E2** is confirmed by empirical results. For both generated and gold headlines of the test set we observe better performances when the classifier is trained on train-M, which is more similar than train-D to the test set, in terms of controlling for the topic, (settings1-3). We also see a gap between validation and test performance in all settings, but smaller when the classifier is trained on train-M (**E3**).

Lastly, there is a drop in performance between in-generated and cross-generated headlines for both setting1-3 and setting2-4, although the performance on cross generated headline is still higher than the random baseline (matching **E4**). This goes to show that when a model trained on *la Repubblica* is asked to generate a headline starting from an *Il Giornale* article, it will do so preserving the style it has learnt from *la Repubblica*, in spite of having generated from the other newspaper's text.

As final evidence, we trained a newspaper-agnostic generator by mixing half of *La Repubblica* and half of *Il Giornale* from train-M (weakly aligned, closer to the test set than train-D), with a resulting size comparable to the other training sets (65k). By design, this model cannot learn any newspaper-specific style, and we, therefore, expect it to be unable to produce any newspaper-specific traits in generation. The measurable consequence of this is that the classifier should indeed not be able to distinguish them. A resulting average F1 score of 0.47, when compared to the scores in Table 6.9, is further proof that our models are indeed learning newspaper-specific style for headline generation.

For completeness, and to give an idea of the generated headlines we obtain using the various models, we report a few examples in Table 6.11. This shows two examples of headlines (one from *la Repubblica* and one from *Il Giornale*) with the automatically generated headlines versions in the different settings.

6.4.5 Conclusions

We trained a few pointer network models under different training settings that learnt to generate headlines according to a given newspaper's style, controlling for topic biases. We also trained a few classifiers that are able to distinguish the source of a given headline with high accuracy. Using such classification models as evaluators we were able to verify that the generators we have trained are indeed style-aware. This was confirmed through an additional experiment that showed that if the headlines are generated by a model trained in a newspaper-agnostic fashion, the classifier is indeed not able to distinguish them.

This whole evaluation procedure is done in a completely automated fashion. This is an advantage not only in terms of saving human effort but especially because our experiments suggest that humans cannot perform this task reliably enough. An aspect to concentrate on in future work concerns the nature of the human judges who perform the evaluation. It would be desirable to collaborate with journalists, possibly title-creating experts from the specific newspapers we work with. Such experts should be better able than lay people to spot and judge whether the generated style is appropriate for their own newspaper. At earlier stages of this work, we did seek collaboration with one of the two papers we worked but received a negative response. We still find this would be a valuable avenue to explore, and we plan to do it in the future. In any case, coupling generation and classification appear

to be a successful evaluation methodology which we believe can be applied more generally, especially in absence of reliable human judgement.

6.5 On the interaction of automatic evaluation and task framing in headline style transfer

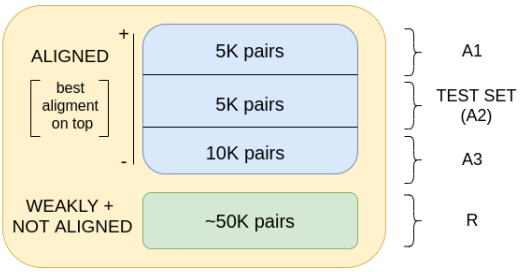
The evaluation of Natural Language Generation (NLG) systems is intrinsically complex. This is in part due to the virtually open-ended range of possible ways of expressing content, making it difficult to determine a 'gold standard' or 'ground truth'. As a result, there has been growing scepticism in the field surrounding the validity of corpus-based metrics, primarily because of their weak or highly variable correlations with human judgments [254, 251, 250, 45]. Human evaluation is generally viewed as the most desirable method to assess generated text [225, 174]. In their recent comprehensive survey on the evaluation of NLG systems, [45] stress that it is important that any used untrained automatic measure (such as BLEU, ROUGE, METEOR, etc) correlates well with human judgements.

At the same time, the human evaluation also presents its challenges and there have been calls for the development of new, more reliable metrics [226]. Beyond the costs associated with using humans in the loop during development, it also appears that certain linguistic judgment tasks are hard for humans to perform reliably. For instance, human judges show relatively low agreement in the presence of syntactic variation [41]. By the same token, [85] observe at best moderate correlations between human raters on stylistic dimensions such as politeness, colloquialism and naturalness.

In [79] we presented three independent judges with headlines from two Italian newspapers with distinct ideological leanings and in-house editorial styles. When asked to classify the headlines according to which newspaper they thought they came from, all three annotators performed the task with low accuracy (ranging from 57% to 62%). Furthermore, agreement was very low (Krippendorff's $\alpha = 0.16$). The agreement was similarly low on classifying automatically generated headlines ($\alpha = 0.13$ or 0.14 for two different generation settings). These results suggest that human evaluation is not viable, or at least not sufficient, for this task.

In [75] we focus on the same style-transfer task using headlines from newspapers in Italian, but address the question of whether a series of classifiers that monitor both style strength as well as content preservation, the core aspects of style transfer [102, 214, 193], can shed light on differences between models.

We also add some untrained automatic metrics for evaluation. As observed above, the fact that humans cannot perform this task reliably makes it impossible to choose such metrics based on good correlations with human judgement [45]. Therefore, relying on previous work, we compare the insights gained from our classifiers with those obtained from BLEU [231] and ROUGE [181], since they are commonly used metrics to assess performance for content preservation and summarisation. Other common metrics such as METEOR [172] and BLEURT [273], which in principle would be desirable to use, are not applicable to our use case as they require resources not available for Italian.



(a) Data splits

training sets		
EVAL	main	R+A3+A1
	HH	A1 + random pairs
	AH	R+A3+A1
TASK	SUM	R+A3
	S2S1	A3 (10K)
	S2S2	A3+A1 (15K)
	S2S3	A1 (5K)

(b) Training sets

FIGURE 6.13: Data splits and their use in the different training sets

More specifically, we train a classifier which, given a headline coming from one of two newspapers with distinct ideological leanings and in-house styles, can identify the provenance of the headline with high accuracy. We use this (the 'main' classifier) to evaluate the success of a model in regenerating a headline from one newspaper, in the style of the other. We add two further consistency checks, both of which aim at content assessment, and are carried out using additional classifiers trained for the purpose: (a) a model's output headline should still be compatible in content with the original headline; (b) the output headline should also be compatible in content with the article to which it pertains. A headline is deemed to be (re)generated successfully in a different style if both (a) and (b) are satisfied, and the main classifier's decision as to its provenance should be reversed, relative to its decision on the original headline.

A core element in our setup is testing our evaluation classifiers/strategies in different scenarios that arise from different ways of framing the style transfer task, and different degrees of data availability. Indeed, we frame the task either as a translation problem, where a headline is rewritten in the target style or as a summarisation problem, where the target headline is generated starting from the source article, using a summarization model trained on target style. The two settings differ in their needs in terms of training data as well as in their ability to perform the two core aspects of style transfer (style strength and content preservation).

We observe how evaluation is affected by the different settings, and how this should be taken into account when deciding what the best model is.

Data and code are available at `https://github.com/michelecafagna26/CHANGE-IT`. The data and task settings also lend themselves well as material for a shared task, and they have indeed been used, with the summarization system described here as a baseline, in the context of the EVALITA 2020 campaign for Italian NLP [77].

6.5.1 Task and Data

Our style transfer task can be seen as a "headline translation" problem. Given a collection of headlines from two newspapers at opposite ends of the political spectrum, the task is to

change all rightwing headlines to headlines with a leftwing style, and all leftwing headlines to headlines with a rightwing style, while preserving content. We focus on Italian in this contribution, but the methodology we propose is obviously applicable to any language for which data is available.

Collection We used the dataset introduced in 6.2.1. In this work, we balanced across the two sources by performing undersampling. Though we are concerned with headlines, full articles are used in two ways: (a) *alignment*; and (b) the consistency check classifiers (see Section 6.5.3 for details). For the former, we leverage the alignment procedure proposed by [39] and we split our dataset into strongly aligned, weakly aligned and non-aligned news. The purpose of alignment is to control for potential topic biases in the two newspapers so as to better disentangle newspaper-specific style. Additionally, this information is useful in the creation of our datasets, specifically as it addresses the need for parallel data for our evaluation classifiers and the translation-based model (see below).

Data splitting We split the dataset into *strongly aligned news*, which are selected using the stricter threshold (~20K aligned pairs), and *weakly aligned and non-aligned news* (~100K article-headline pairs equally distributed among the two newspapers). The aligned data is further split as shown in Figure 6.13. `SA` is left aside and used as a test set for the final style transfer task. The remaining three sets are used for training the evaluation classifiers and the models for the target task in various combinations. These are described in Figure 6.13 and in connection with the systems' descriptions.[11]

6.5.2 Systems

Our focus is on the interaction of different evaluation settings and approaches to the task. Accordingly, we develop two different frameworks with different takes on the same problem: (a) as a true translation task, where given a headline in one style, the model learns to generate a new headline in the target style; (b) as a summarization task, where headlines are viewed as an extreme case of summarization and generated from the article. We exploit article-headline generators trained on opposite sources to do the transfer. This approach does not in principle require parallel data for training.

For the translation approach (`S2S`), we train a supervised BiLSTM sequence-to-sequence model with attention from OpenNMT [159] to map the headline from left-wing to right-wing, and vice-versa. Since the model needs parallel data, we exploit the aligned headlines for training. We experiment with three differently composed training sets, varying not only in size but also in the strength of the alignment, as shown in Figure 6.13.

For the summarization approach (`SUM`), we use two pointer-generator networks [272], which include a *pointing mechanism* able to copy words from the source as well as pick them from a fixed vocabulary. This allows for a better handling of out-of-vocabulary words, rendering generation more accurate. One model is trained on the *la Repubblica* portion of

[11] Note that all sets also always contain the headlines' respective full articles, though these are not necessarily used.

the training set, the other on *Il Giornale*. In a style transfer setting we use these models as follows: Given a headline from *Il Giornale*, for example, the model trained on *la Repubblica* can be run over the corresponding article from *Il Giornale* to generate a headline in the style of *la Repubblica*, and vice versa. To train the models we use subset `R`, but we also include the lower end of the aligned pairs (`A3`), see Figure 6.13.

6.5.3 Evaluation

Our fully automatic strategy is based on a series of classifiers to assess style strength and content preservation. For style, we train a single classifier (*main*). For content, we train two classifiers that perform two 'consistency checks': one ensures that the two headlines (original and transformed) are still compatible (*HH classifier*); the other ensures that the headline is still compatible with the original article (*AH classifier*). See also Figure **??**.

In what follows we describe these classifiers in more detail. When discussing results, we will show how the contribution of each classifier is crucial towards a comprehensive evaluation.

Main classifier The main classifier uses a pre-trained BERT encoder with a linear classifier on top fine-tuned with a batch size of 256 and sequences truncated at 32 tokens for 6 epochs with learning rate 1e-05. Given a headline, this classifier can distinguish the two sources with an f-score of approximately 80% (see Table 6.12). Since style transfer is deemed successful if the original style is lost in favour of the target style, we use this classifier to assess how many times a style transfer system manages to reverse the main classifier's decisions.

HH classifier This classifier checks compatibility between the original and the generated headline. We use the same architecture as for the main classifier with a slightly different configuration: max. sequence length of 64 tokens, batch size of 128 for 2 epochs (early-stopped), with learning rate 1e-05. Being trained on strictly aligned data as positive instances (`A1`), with a corresponding amount of random pairs as negative instances, it should learn whether two headlines describe the same content or not. Performance on gold data is .96 (Table 6.12).

AH classifier This classifier performs yet another content-related check. It takes a headline and its corresponding article and tells whether the headline is appropriate for the article.The classifier is trained on article-headline pairs from both the strongly aligned and the weakly and non-aligned instances (`R+A3+A1`, Figure 6.13). At test time, the generated headline is checked for compatibility against the source article. We use the same base model as for the main and HH classifiers with a batch size of 8, same learning rate and 6 epochs. Performance on gold data is >.97 (Table 6.12).

Overall compliancy We calculate a compliancy score which assesses the proportion of times the following three outcomes are successful (i) the *HH classifier* predicts 'match'; (ii) the *AH classifier* predicts 'match'; (iii) the *main classifier*'s decision is *reversed*. As upper

		prec	rec	f-score
main	**rep**	0.77	0.83	0.80
	gio	0.84	0.78	0.81
HH	**match**	0.98	0.95	0.96
	no match	0.95	0.98	0.96
AH	**match**	0.96	0.99	0.98
	no match	0.99	0.96	0.97

TABLE 6.12: Performance of the classifiers on gold data.

		HH	AH	Main	Compl.	BLEU	ROUGE
lightgray	without top aligned data						
SUM	**rep2gio**	.649	.876	.799	.449	.020	.145
	gio2rep	.639	.871	.435	.240	.026	.156
	avg	**.644**	**.874**	.616	.345	**.023**	**.151**
S2S1	**rep2gio**	.632	.842	.815	.436	.011	.136
	gio2rep	.444	.846	.864	.321	.012	.130
	avg	.538	.844	**.840**	**.379**	.012	.133
lightgray	with top aligned data						
S2S2	**rep2gio**	.860	.845	.845	.549	.018	.159
	gio2rep	.612	.846	.847	.442	.016	.151
	avg	.736	**.846**	**.849**	**.496**	**.017**	**.155**
S2S3	**rep2gio**	.728	.844	.845	.520	.012	.139
	gio2rep	.760	.848	.649	.420	.013	.156
	avg	**.744**	**.846**	.747	.470	.013	.148

TABLE 6.13: Performance on test data.

bound, we find the compatibility score for gold at 74.3% for transfer from *La Repubblica* to *Il Giornale* (*rep*2*gio*), and 78.1% for the opposite direction (*gio*2*rep*).

6.5.4 Results and Discussion

Table 6.13 reports results of our evaluation methods both for the summarization system (SUM) and for the style transfer systems (S2S) in the different training set scenarios.

The top panel in Table 6.13 shows the results for systems where training data is weakly aligned or unaligned. The summarization system SUM does better at content preservation (HH and AH) than S2S1. However, its scores on the *main* classifier are worse in both transfer directions, as well as on average. The average compliance score is higher for S2S1. In summary, for data that is not strongly aligned, our methods suggest that style transfer is better when conceived as a translation task. BLEU is higher for SUM, but the overall extremely low scores across the board suggest that it might not be a very informative metric for this setup, although commonly used to assess content preservation in style transfer [246]. Our HH and AH classifiers appear more indicative in this respect, and ROUGE scores seem to correlate a

bit more with them when compared to BLEU. It remains to be investigated whether BLEU, ROUGE, and our content-checking classifiers do in fact measure something similar or not.

With better-aligned data (bottom panel), the picture is more nuanced. Here, the main comparison is between two systems trained on strongly aligned data, one of which (`S2S2`) has additional, weakly aligned data. The overall compliance score suggests that this improves style transfer (and this system is also the top-performing one overall, also outperforming `S2S1` and `SUM`). As for content preservation (AH and HH scores), `S2S3` is marginally better on average for HH, but not for AH, where the two systems are tied.

Overall, the results of the classification-based evaluation also highlight a difference between a summarisation-based system (`SUM`), which tends to be better at content preservation, compared to a translation-based style transfer setup (especially `S2S2`) which transfers style better. Clearly, a corpus-based metric such as BLEU fails to capture these distinctions, but here does not appear informative even just for assessing content preservation.

One aspect that will require further investigation, since we do not have a clear explanation for it as of now, is the performance difference between the two translation directions. Indeed, transforming a *La Repubblica* headline into an *Il Giornale* headline appears more difficult than transforming headlines in the opposite directions, under most settings.

6.5.5 Conclusions

This paper addressed the issue of how to evaluate style transfer. We explicitly compared systems in terms of the extent to which they preserve content, and their success at transferring style. The latter is known to be hard for humans to evaluate [85, 79]. Our aim was primarily to see to what extent different evaluation strategies based on purposely trained classifiers could distinguish between models, insofar as they perform better at either of these tasks and in different training scenarios.

Our findings suggest that our proposed combination of classifiers focused on both content and style transfer can potentially help to distinguish models in terms of their strengths. Interestingly, a commonly used metric such as BLEU does not seem to be informative in our experiments, not even for the content preservation aspects.

To the extent that stylistic distinctions remain hard for humans to evaluate in setups such as the one used here, a classification-based approach with consistency checks for content preservation is a promising way forward, especially to support development in a relatively cheap and effective way.

Future work will have to determine how the various metrics we have used relate to each other (especially our classifiers and BLEU/ROUGE), and whether the human judgement can be successfully brought back, and in the case in what form, at some stage of the evaluation process.

6.6 Final remarks

In this chapter several studies about news headlines have been described. Those studies tackled stylistic aspects of two Italian newspaper on the opposite side of the political spectrum: *La Repubblica* (left-wing) and *Il Giornale* (right-wing). The machine learning models adopted are able to distinguish the headlines from the two newspapers with higher accuracy than humans. This lets us apply those model to evaluate the capabilities of NLG systems of generating headlines compliant with the style of a specific newspaper. Moreover, similar machine learning techniques have been successfully applied to evaluate the content preservation capabilities of the NLG systems. In this settings, we have been able to create an evaluation framework to evaluate both stylistic and content aspects of the NLG systems. However, thanks to human evaluation, turned out that human-produced headlines are much more attractive than generated ones. This fact gives us space for further works which will focus on improving NLG systems in producing more engaging texts and in study how machine learning methods may model engagement aspects.

Chapter 7

Carving Italian into a Language Model

7.1 Introduction

Language Models (LMs) based on pre-trained architectures such as BERT [86] and GPT-2 [243] have provided impressive improvements across several NLP tasks. While for BERT-based architectures several monolingual models other than English have been developed, language-specific implementations of generative pre-trained transformer-based models, such as GPT-2, are not widely available yet. As a contribution to fill this gap, we developed `GePpeTto`, the first generative language model for Italian, using the original GPT-2 as a blueprint [78].

For the evaluation of the model we adopt here an encompassing approach, performing both automatic and human-based evaluations. The automatic assessment consists of two strategies: the first involves calculating perplexity across different language models trained on various datasets representing different genres. This serves to understand how good `GePpeTto` is as a language model, and how much it captures the various genres. The second one is a profiling analysis where, by means of a series of linguistic features, we capture some of `GePpeTto`'s writing characteristics and compare them to those of the data it was trained on. Finally, human evaluation is performed over a sentence completion task where `GePpeTto` is evaluated against gold standard sentences as well as a simple Markov-based baseline.

We make the model available to the community: `https://github.com/LoreDema/GePpeTto`.

7.2 `GePpeTto`

`GePpeTto` was trained using the original settings of GPT-2 on a collection of Italian texts amounting to almost 13GB. Details on data and model's parameters are provided in the following sections.

7.2.1 Data

The training set comprises two main sources. The first one is a dump of Italian Wikipedia (November 2019), consisting of 2.8GB of text. The content was extracted using the Wikiextractor tool [5]. The second one is ItWac corpus [15], which amounts to 11GB of web texts. This collection provides a mix of standard and less standard Italian, on a rather wide chronological span, with older texts than the Wikipedia dump (the latter stretches only to the late 2000s).

Minimal processing was applied to the texts. All Wikipedia documents were prefixed by the token “Wikipedia" followed by the page’s title words. All ItWac texts were introduced by the token “Links" followed by the webpage address the text was coming from. For all texts in both collections, the end of the documents was marked with the string <|endoftext|>, as done for the original GPT-2 training set [243].

7.2.2 Model

GePpeTto was trained on 4 GPUs Tesla T4, using the PyTorch implementation available on huggingface.[1] The model’s size corresponds to GPT-2 small, with 12 layers and 117M parameters [243]. The vocabulary size is 30,000, the batch size is 32 and the block size is 100. We use 10,000 warmup steps, the adam optimiser, and an initial learning rate of 5e-5. We train the model for 620,000 steps.

7.2.3 Examples

Through some selected examples, we illustrate here GePpeTto’s generation behaviour. We prompt it with different inputs and show how the generated text is indeed style-compliant with the prompts. In the first example, we start the prompt with the ‘keyword’ “Wikipedia", priming GePpeTto to reproduce the text in the style and format of Wikipedia pages (recall that the tag “Wikipedia" was proposed to all Wikipedia training data).

prompt: “Wikipedia La Repubblica del Burmini"

“Wikipedia La Repubblica del Burmini \n\nLa Repubblica del Burmini (in francese “La Repubblica del Burginais"), originariamente repubblica di Francia, venne creata nel 1859 insieme al Regno d’Italia, con capitale Bourges, per promuovere l’unificazione dei due territori della Francia. Le città della Francia si unirono nel 1877 ed i cinque stati facenti parte dell’impero erano Francia, Italia e Savoia."

In the second example, we prompt GePpeTto with the same text “*La Repubblica del Burmini*" but preposing to it the ‘keyword’ “Links" followed by the web address of an Italian newspaper, priming GePpeTto to reproduce a text in the style of a news article. Interestingly the output is in news style but the prompt is interpreted as the header of the news, i.e., ‘place and date’.

[1] https://huggingface.co.

prompt: “Links http://www.corriere.it La Repubblica del Burmini"

“Links http://www.corriere.it La Repubblica del Burmini - 14 gennaio 2003. Nota: Per gli ambiti regionali, le Regioni dovranno determinare in un solo anno le modalità di trasporto, tra gli altri, degli studenti del secondo ciclo, e le modalità per pagare eventuali supplementi a carico del servizio sanitario nazionale. Si tratta, in altre parole, di un decreto legislativo con un semplice riferimento alla Regione Lombardia.

In the following three examples we can see that even without using any specific keywords, `GePpeTto`'s is easily style-primed by clear prompts. In the first case, we use as input a string prompting forum-like language, in the second one, we use legal terms in the prompt, and in the last one, we prompt `GePpeTto` with a more sophisticated language, which the model appears to readily pick up in its generated text.

prompt: “Piero DM wrote:"

“Piero DM wrote: >Una domanda sul lavoro a tempo pieno dovrebbe far sorridere.....<< Scusate, ma vorrei avere una vostra opinione."

prompt: “Visto l’articolo 4 del dpcm:"

“Visto l’articolo 4 del dpcm 29 dicembre 1999, con il quale viene disposto l’inquadramento tra il personale interessato alle operazioni e le RSU per l’applicazione del quarto comma del predetto articolo 4, paragrafo 1, lettera a) del presente accordo, nei ruoli dei giornalisti, delle altre amministrazioni dello Stato, delle agenzie ed aziende autonome; Vista la legge 29 marzo 1982, n. 93, e successive modificazioni,

prompt: “l’intricato esistenzialismo"

“l’intricato esistenzialismo diventa il motore di quella riflessione metafisico-filosofica di cui il primo pensatore si avvale. L’altruismo di fondo appartiene quasi completamente alla filosofia; l’acuto verso la razionalità senza limite.

7.3 Automatic Evaluation

`GePpeTto` is trained as a language model for Italian. To assess its closeness to actual Italian texts, we calculate perplexity on a variety of sources, including a small leave out test set (1%) of `GePpeTto` s training corpus (Section 7.3.1). In addition, we explore more in-depth `GePpeTto`'s linguistic profile by comparing its production with human-written texts along with a series of linguistic features (Section 7.3.2).

7.3.1 Perplexity

As a first evaluation, we are interested in understanding the quality of `GePpeTto` as a language model in its own training domain. As a second evaluation, we want to test its performance at

zero-shot domain transfer (i.e. language modelling of a different domain). We use perplexity as a measure of language modelling performance. The different domains we consider, and the relative corpora we use, are as follows:

- own domains: Wikipedia and ItWac;
- legal domain: a corpus of Italian laws scraped from EUR-Lex[2] (tables excluded);
- news: a corpus of articles from the online versions of two major Italian newspapers, namely *la Repubblica*[3] and *Il Giornale*[4] [79];
- social media: a corpus of forum comments [203].

Perplexity scores are reported in Table 7.1. As we could expect, `GePpeTto` performs better on its own domains, with Wikipedia being the best of the two. Although ItWac is four times bigger than Wikipedia, the lower performance on the former might be due to the fact this corpus is open domain with a large diversity of styles, while Wikipedia is more 'standardised'. Consistently with this hypothesis, we observe a similar trend in 'out-of-domain' testing, where `GePpeTto` performs better on domains with a well-coded style, namely legal documents. On domains with less coded styles, such as news and especially forum comments, we observe a drop in performance.

If we compare perplexity scores with the original English GPT-2 small model, we see that `GePpeTto`'s results are slightly worse on the own domain corpora, which could be due to the smaller size of the training set. Out-of-domain perplexity scores are comparable between the two models.

DOMAIN	PERPLEXITY
Wikipedia	26.1052
ItWac	30.3965
Legal	37.2197
News	45.3859
Social Media	84.6408

TABLE 7.1: Perplexity of `GePpeTto` over several in-domain and out-of-domain corpora.

7.3.2 Linguistic Profiling

For our second evaluation, we used Profiling-UD [38], a tool for the automatic analysis of texts that extracts several linguistic features of varying complexity. These features range from raw text properties, such as average length of words and sentences, to lexical, morpho-syntactic, and syntactic properties, such as part-of-speech (POS) distribution and inflectional properties of verbs. More complex aspects of sentence structure are derived from syntactic

[2] https://eur-lex.europa.eu/
[3] https://www.repubblica.it
[4] https://www.ilgiornale.it/

	Original		GePpeTto	
Feature	μ	std	μ	std
CPT	4.809	0.959	4.750	1.127
TPS	32.302	28.322	20.382	11.127
TPC	12.393	11.504	10.711	8.529
LL_{max}	13.290	13.370	8.922	6.112
LL_{avg}	2.555	1.002	2.373	0.676

TABLE 7.2: Main linguistic features considered in our analysis. CPT = chars per token, TPS = token per sentence, TPC = tokens per clause, LL = links length.

annotation, and model global and local properties of the parsed tree structure, such as the order of subjects/objects with respect to the verb, the distribution of syntactic relations, and the use of subordination.

In our analysis, we focus on two macro aspects of GePpeTto's output, namely lexical complexity and syntactic complexity, and compare them to human productions. To do so, we rely on a selection of Profiling-UD's features which we use as proxies for the macro-aspects that we consider.

We run the profiling analysis on a sample of both gold and generated texts. For gold, we randomly sample the test set for a total of about 19k sentences. For GePpeTto, we picked the first token from each of the 19k gold sentences and used it as a prompt to the model. These are the generated texts that we profile.

Lexical complexity. We proxy lexical complexity with the number of characters per word, the overall frequency of tokens, also with reference to an external dictionary, and POS distribution.

The number of characters per token (CPT), which indicates whether shorter (usually more common) or longer (usually more complex/specialised) words are used, is completely comparable across the original (4.80, std=0.96) and GePpeTto's (4.75, std=1.13) language models – see Table 7.2. This suggests that the complexity of the used vocabulary is not that different.

We compute a reference dictionary of token frequency on ItWac (≈1.5 billion tokens), and compare observed token frequency in both gold and generated text to this reference. We observe that in gold sentences, each token has a probability of 0.912 to be in the top 5‰ of most frequent tokens. In the generated sentences, the probability grows to 0.935, suggesting that GePpeTto is more likely to use more frequent words rather than rarer ones. This observation is in line with previous research which showed that for Nucleus Sampled texts, such as those produced by GPT-2, all tokens come from the top-p%, since the long tail is cut off, while for human-produced texts, the probability of all tokens being drawn from the top-p% of the language distribution goes to zero as document length increases [107, 313].

Regarding POS distribution, we observe that while for most POS tags usage is comparable, for a few others the two language models differ. The latter are, specifically, auxiliaries and

	Original		GePpeTto	
POS	μ	std	μ	std
AUX	0.032	0.041	0.040	0.051
PROPN	0.070	0.105	0.081	0.125
PUNCT	0.148	0.103	0.153	0.105
DET	0.140	0.071	0.143	0.078
NUM	0.031	0.072	0.032	0.064
ADP	0.139	0.070	0.138	0.077
PRON	0.037	0.053	0.036	0.058
SCONJ	0.008	0.020	0.008	0.023
NOUN	0.179	0.082	0.172	0.087
VERB	0.079	0.059	0.075	0.065
ADV	0.042	0.060	0.039	0.063
CCONJ	0.027	0.034	0.024	0.037
ADJ	0.063	0.058	0.055	0.062

TABLE 7.3: POS considered in our analysis.

proper nouns, which GePpeTto tends to overgenerate in comparison to the original model, and adjectives, which GePpeTto instead uses less than in the original texts. This is observable also for nouns and verbs, but the differences are relatively minimal. Conjunctions are also overall less frequent in GePpeTto. See Table 7.3 for details.

Syntactic complexity. At the level of syntax, we proxy complexity by the number of tokens per sentence, and the number of tokens per clause. We also look at the length of a dependency link, which is calculated as the number of words occurring linearly between the syntactic head and its dependent (excluding punctuation dependencies). The value associated with this feature corresponds to the average value extracted for all dependencies in a text. This information is complemented with the feature *Maximum dependency link* corresponding to the longest dependency link for each sentence.

When comparing the number of tokens per sentence (TPS, Table 7.2), we see that it's much lower for GePpeTto's production rather than for human texts (20.4 tokens per sentence on average for GePpeTto vs 32.3 for gold texts), indicating that GePpeTto generates shorter sentences. Contextually, we also observe that GePpeTto's generated sentences exhibit less variation in length (smaller STD) than human sentences (larger STD).

The difference in the number of tokens at the clause level is relatively smaller, with clauses of length 12.4 in human texts vs 10.7 in GePpeTto (TPC, see Table 7.2). Considering that a clause is proxied by the presence of a verbal/copular head, it seems that sentences produced by GePpeTto, though shorter, are similar in complexity given the proportional distribution of verbal heads.

The above values taken together might suggest that while complexity at the macro-level (sentence length) is higher for natural sentences, at the micro-level (clause length) the complexity of GePpeTto's generations and human texts is more similar. While this intuition will require further linguistic analysis, it seems to be confirmed by the data we have if we

look at the length of syntactic links. This feature proxies quite well syntactic complexity since it indicates how maximally far (and how far on average) a dependent and its head are within a sentence. Both the maximum length and the average length are higher for human texts (LL_{max} and LL_{avg}, see Table 7.2). However, if we look at them proportionally to sentence length, we find that they are absolutely comparable: normalising the longest link by the number of tokens per sentence (LL_{max}/TPS), we obtain basically the same value for gold (0.411) and for `GePpeTto` (0.438). This suggests that `GePpeTto` produces somewhat shorter sentences, but their internal complexity relatively corresponds to the internal complexity of the longer sentences produced by humans.

7.4 Human evaluation

We also test `GePpeTto`'s ability to generate Italian texts through a sentence completion task. The automatically generated sentences are presented to human subjects for evaluation on perceived naturalness and compared to gold ones and to a baseline.

While the original (gold) texts represent an upper bound for `GePpeTto`, we do not actually have a lower bound against which the quality of `GePpeTto` can be assessed. To provide a comparison, we train a simple Markov model that would be able to generate text and use it as our baseline. Since the size of a Markov model dramatically grows with its vocabulary size, we use 1 million randomly sampled sentences from the same training-set used for `GePpeTto`. We train a Markov chain generator using the `markovify`[5] implementation with state size 2, then we generate synthetic texts starting from the last 2 tokens of the same prompts used for `GePpeTto`.

7.4.1 Tasks

Human subjects are asked to perform two evaluation tasks. One is a comparative ranking task, where subjects are asked to rank three portions of text (produced by gold, `GePpeTto`, baseline) according to perceived naturalness. The other is a classification task, where subjects are asked to tell, according to their intuition, if a portion of text, seen in isolation, is automatically generated (*yes, no, can't tell*).

Experimental design. The experiment includes 12 conditions of the stimulus material in a 4x3 design. One level (A) with three conditions is given by {gold,`GePpeTto`, baseline}. The second level (B) is the `prompt+completion` combination that results in 4 conditions {5+5, 5+10, 10+5, 10+10}. We use 100 different prompts (randomly selected gold sentences truncated at 5 and 10 tokens). Each of the 100 prompts enters each of the 12 conditions of the 4x3 design, for a total of 12 different stimuli. Basically, each 5 or 10 tokens prompt is completed with 5 or 10 tokens coming either from gold, `GePpeTto`, or the baseline model. Table 7.4 shows an example of all the stimuli deriving from the same 5- or 10-token prompt.

Each subject is assigned either to the ranking or to the classification task.

[5] https://github.com/jsvine/markovify.

5 token prompt: Mentre per quanto riguarda gli	
10 token prompt: Mentre per quanto riguarda gli accordi per la fornitura di	
	Gold
5+5	Mentre per quanto riguarda gli accordi per la fornitura di
5+10	Mentre per quanto riguarda gli accordi per la fornitura di latte, in scadenza questa
10+5	Mentre per quanto riguarda gli accordi per la fornitura di latte, in scadenza questa
10+10	Mentre per quanto riguarda gli accordi per la fornitura di latte, in scadenza questa settimana, Alemanno ha detto
	`GePpeTto`
5+5	Mentre per quanto riguarda gli emendamenti, fa presente che il
5+10	Mentre per quanto riguarda gli emendamenti, fa presente che il suo gruppo non ha sottoscritto
10+5	Mentre per quanto riguarda gli accordi per la fornitura di beni e servizi, i fatti
10+10	Mentre per quanto riguarda gli accordi per la fornitura di beni e servizi, i fatti in suo possesso hanno come
	Markov-based baseline
5+5	Mentre per quanto riguarda gli aspetti più significativi del mondo
5+10	Mentre per quanto riguarda gli aspetti più significativi del mondo editoriali, con priorità di sviluppo
10+5	Mentre per quanto riguarda gli accordi per la fornitura di biciclette elettriche a 48 bit
10+10	Mentre per quanto riguarda gli accordi per la fornitura di biciclette elettriche a 48 bit (281,5 trilioni di operazioni e

TABLE 7.4: Example outputs (stimuli) for different prompt lengths of the same original sentence.

In the ranking, we opt for a between-subject evaluation set up by assigning each subject to one of the (B) conditions and offer the three versions of (A) to be ranked. For example, one subject is asked to evaluate all the 100 prompts in the 5+5 configuration (dimension B) for the three realisations, i.e., gold, `GePpeTto`, and baseline (dimension A).

For the classification experiments, we again opt for a between-subject evaluation set up, this time by assigning each subject to one of the 12 conditions, randomly picked up for each prompt. In other words, we make sure that each subject is exposed to only one completion per prompt, randomising prompt order. By seeing only one (out of 12) realisation per prompt, each subject sees a given prompt only once and we can therefore avoid cross-comparison effects of different completions of the same prompt, which could otherwise potentially lead again to an implicit ranking task.

Material. The materials are prepared as follows: we have selected 100 random documents/sentences and have cut them at their 5 first tokens and also their 10 first tokens. Each

5-token and 10-token prompt was given to `GePpeTto` and baseline so that the models could continue the text.

For each prompt, we obtain one single generated text by the two automatic models and chop them at 5 or at 10 tokens. In other words, each chopped version is derived from the same generated output which is just cut at different lengths.

We cut the sentences (including the original one) to control for the effect of text length. Indeed, we observed in Section 7.3.2 that `GePpeTto` generates shorter sentences than humans, which could represent a strong bias in evaluation. In Table 7.4, we show examples of all the possible stimulus material configurations according to the `prompt+completion` conditions of level (B).

Instructions and subjects. For both the ranking and classification experiments, subjects were told that they will have to evaluate excerpts of text along a 'more natural vs. more artificial' dimension. All stimuli used in both scenarios are the same.

For the ranking scenario, subjects were asked to "*rank the given examples from the most natural to the most artificial"*, where the inputs are three texts (gold, `GePpeTto`, baseline), all starting with the same prompt, thus the same five or ten tokens.

For the classification scenario, subjects saw instead the portions of text in isolation, and could answer *yes*, *no*, or *can't tell* to the question "*according to your intuition is this sentence written by an artificial intelligence?*".

A total of 24 unique subjects (12 females) carried out the tasks using Google Forms (see Figure 7.1 for a snapshot of the interfaces.) Twelve subjects (6 females) were assigned to Task 1 and the others to Task 2. Each subject evaluated 100 cases, and each case was evaluated by three different subjects.

7.4.2 Results

First, we discuss the results of our human evaluation separately, with observations related to the ranking task and observations related to the classification task. Subsequently, we knit together the two outcomes to draw a wider picture of how humans assess the quality of `GePpeTto`'s output.

Ranking Overall, results show that the most frequently chosen completion is the gold one, followed by `GePpeTto` and then the Markov baseline, but the baseline is far more distant from `GePpeTto` than `GePpeTto` from gold (Figure 7.2). If we look at results in more detail (see Table 7.5), based on the variable that we have considered in the experimental setup, namely length of input and continuation as well as overall sentence length, we observe that the order of preference for gold is 10+10, then 5+10, then 10+5, and lastly 5+5, while for the automatic models the order is 5+5, 10+5, 5+10, and then 10+10, suggesting the following.

First, the shortest the sentence, the hardest it is to discriminate between gold and generated text; indeed, the 5+5 condition is the one that results best for the two models and worst for gold.

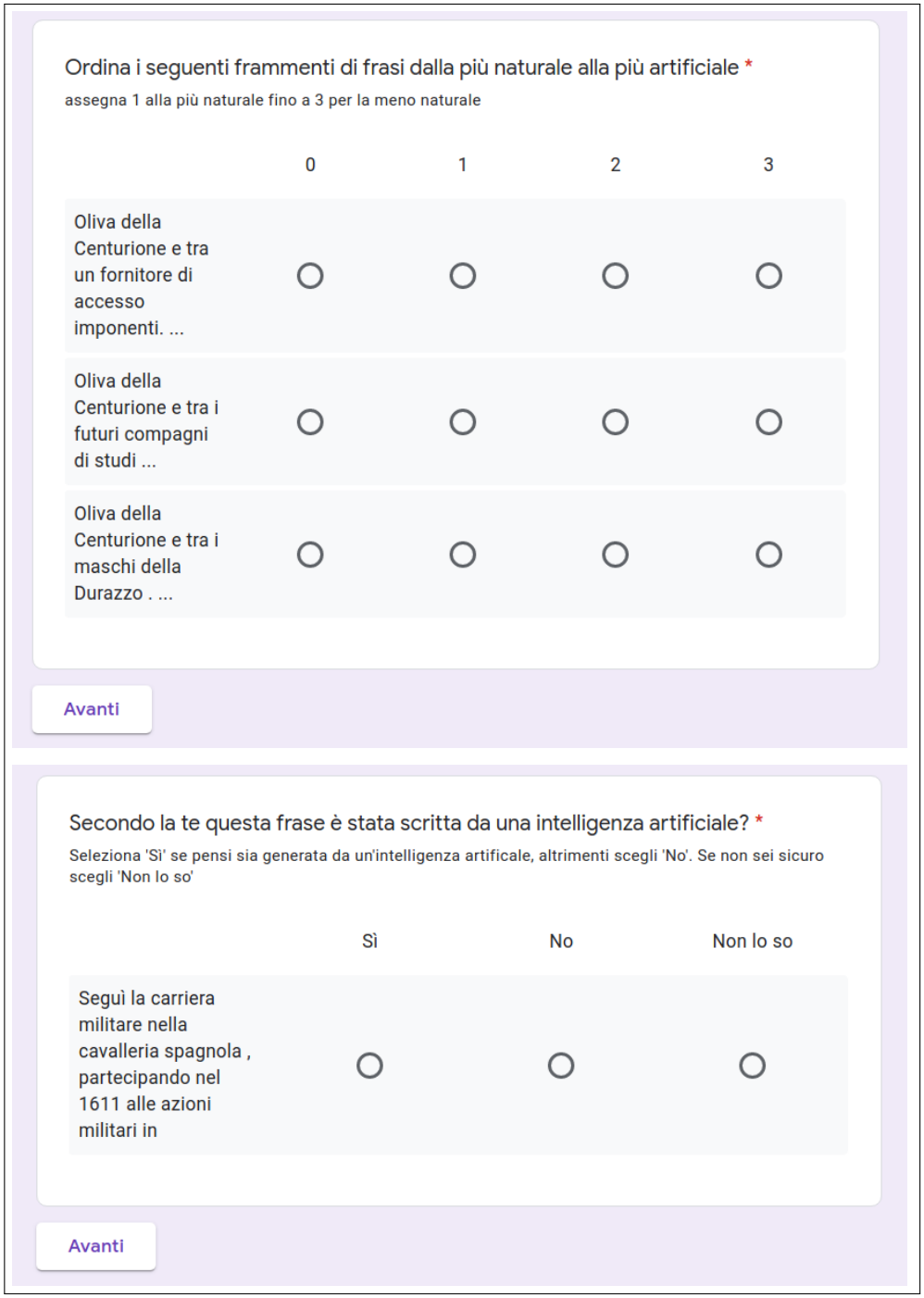


FIGURE 7.1: Annotation interfaces for the ranking and classification tasks.

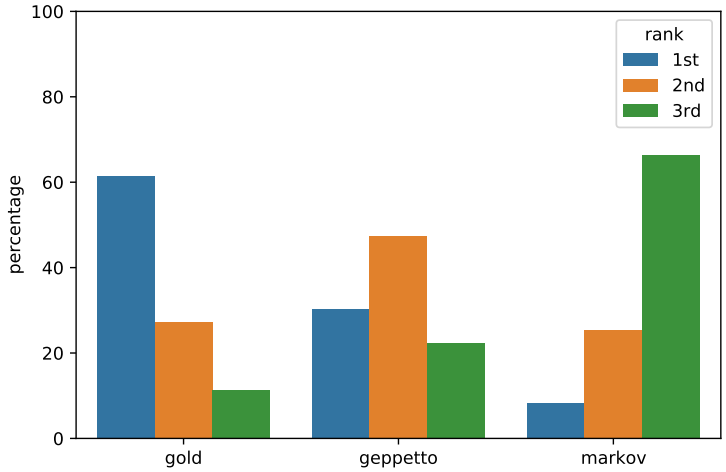


FIGURE 7.2: Ranking results for the three models

model	5+5			5+10			10+5			10+10		
	1^{st}	2^{nd}	3^{rd}	1^{st}	2^{nd}	3^{rd}	1^{st}	2^{nd}	3^{rd}	1^{st}	2^{nd}	3^{rd}
Gold	54	30	16	62	31	7	60	27	13	70	21	9
GePpeTto	34	43	23	30	46	24	33	43	24	23	59	18
Markov	12	27	61	8	23	69	7	30	63	7	20	73

TABLE 7.5: Percentages of ranking results according to the various stimulus material conditions.

Second, when the sentence is the longest (10+10), it is easiest for the subjects to discriminate the gold from the generated sentences. It is also interesting to note that in this condition we observe the largest gap between the two generation models, with GePpeTto getting ranked higher than Markov more than in the other conditions.

Third, at equal sentence length (15 tokens) the situation is a bit fuzzier, but we can observe a slight tendency where it is easier to spot as automatically generated the 5+10 rather than 10+5 cases. This, in combination with the previous observation, seems to imply that the longer the generated text, the easier it is to figure out which texts are automatically produced, which makes sense since there is more 'space' for the models to make mistakes.

Classification Overall, results show that across all conditions, gold sentences are most often rightly identified as not automatically generated (68% of "*no*" to the question of whether the output was produced by artificial intelligence), followed by GePpeTto (54%), and lastly by the Markov baseline (26%), indicating, as expected, that the latter produces the least natural outputs. Figure 7.3 reports the distribution over the various answers. Also in this case the distance between GePpeTto and gold is lower than GePpeTto and the baseline (double in percentage points), indicating that the production of GePpeTto is approaching natural language. It is also interesting to see that the highest percentage of "*can't tell*" is recorded

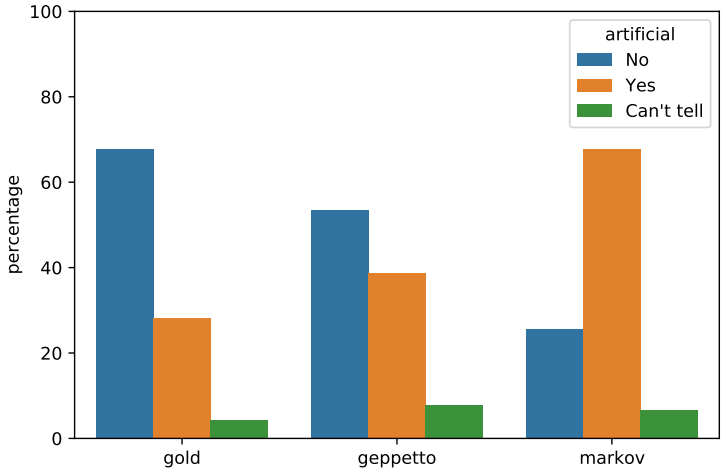


FIGURE 7.3: Classification results for the three models

model	5+5			5+10			10+5			10+10		
	yes	no	ct	yes	no	ct	yes	no	ct	yes	no	ct
Gold	26	66	8	27	68	5	32	63	5	28	71	1
GePpeTto	32	55	13	48	46	6	32	62	6	42	50	8
Markov	62	33	5	80	13	7	61	33	6	71	19	10

TABLE 7.6: Percentages of classification results according to the various stimulus material conditions.
Is the text automatically generated? *{yes, no, can't tell (ct)}*.

for GePpeTto, meaning that for this model it was harder than for baseline and gold to decide whether the text was automatic or not.

Let us look at results in more detail (Table 7.6), focusing again on the length of input and continuation. Regarding continuation, we observe that *+5 conditions are better than *+10 conditions for both automatic models, indicating that the least generated text, the more natural the fragment is perceived.

Regarding input length, we see that for GePpeTto a longer prompt yields better results (10+5 is better than 5+5, and 10+10 is better than 5+10). With 10-token prompts, GePpeTto generates text that is (i) assessed as natural as much as the original text when completed with 5 tokens (62% GePpeTto, 63% original), and (ii) judged as natural 50% of the times when completed with 10 tokens. This seems to suggest that a longer input context is beneficial to GePpeTto when completion size is kept constant. However, we may wonder whether GePpeTto is evaluated as more natural because the generated text is actually better given the more context to start with, or simply because there is more gold text in the stimulus. If it were just for the contribution of a longer gold portion in the stimulus, we should see a similar behaviour for the baseline. Instead, we see that prompt size doesn't matter for the baseline, at least for the 5 token completion case (33% in both 5+5 and 10+5). In the 10-completions (5+10 and 10+10), the larger amount of gold data in the stimulus probably does alleviate a little the very low naturalness induced by the generated text. While we can

tentatively postulate that GePpeTto generates better text when more input is provided, further investigation is required to provide more solid evidence.

Summary of Results. Intersecting the observations from the two experimental setups provides us with a complete picture. In ranking (thus when the models are directly compared), both GePpeTto and the baseline perform best in the 5+5 and 10+5 conditions, suggesting that automatic generation can easily be spotted when compared side by side with human text. In other words, the least generated material, the better.

However, looking at classification, where each textual material is evaluated in isolation, we see that the two models behave in fact very different. First, there is a much larger proportion of cases produced by GePpeTto that are deemed "natural" (54%) compared to Markov (26%). Second, the margin of uncertainty when judging GePpeTto is higher than for the baseline and for the original text. Lastly, given the same completion size, GePpeTto performs better when its prompt is longer. Whether this is an effect of a larger proportion of gold data in the stimulus or it has to do with providing the model with a larger input context is left to future investigation.

7.5 Human Perception in Natural Language Generation

Pre-trained Language Models (PLMs) have proved extremely successful in a variety of NLP task. These models are trained using crawl data which may contain a lot of noise, that is, some of these human data is not perceived so much as human-produced. Indeed, the previous sections have shown that gold sentences are not necessarily assessed as better or more human-sounding than generated texts. On the other hand, there is no clear guidance to tune the model towards the generation that is more human-perceived as far as we know. This also raises the more general, open-discussion issue of what kind of language we expect a language model to have learnt, and thus to generate.

In an ongoing work, we explore such issues by using GePpeTto . We first consider collecting human judgements over texts generated by GePpeTto and human-produced (gold) Italian texts. For a text that could have been generated, but it's perceived as human, or the other way round. We then fine-tune GePpeTto with this data, where the label used is perception, rather than its actual source. Also, inspired by the classifier-based reward used in the style transfer task [169, 116, 194, 268], we further add the reward to the models, to push their classification confidence.

Finally, we study how human-perception-based fine-tuning compares to reward-based fine-tuning, as well as the original model. Specifically, in this work, we conduct both automatic and human-based evaluations. For automatic evaluation, we train a regressor that models perception instead of the actual origin of the text. This serves to produce a robust classifier and provide results that have a higher correlation with human judgments. The human evaluation is performed over a sentence completion task where the gold standard sentence is evaluated against the original GePpeTto, fine-tined GePpeTto, and reward-based GePpeTto.

7.5.1 Data

In order to run our experiments, we need human judgments over a series of gold and generated sentences. The judgements must be elicited according to *perception*: is a given text perceived as generated by a human or a machine? We need these labels to fine-tune our base model towards a model which generates more humanly-perceived texts. We also need labels for test data.

Training Data From the original GePpeTo's training corpus [78], we collected 1400 random gold sentences in the following way. We sentence split all the documents and we picked the first sentence of each document. In order to allow for length variation, which has an impact on perception [78], we selected the first 200 sentences with length 10, 15, 20, 25, 30, 35 and 40 tokens.

To match the gold sentences, we let GePpeTto generate texts starting with prompts consisting of the first word of randomly selected documents. After the texts were generated, we sentence split them and selected the first 200 sentences with length 10, 15, 20, 25, 30, 35 and 40 tokens.

This procedure creates a training set with perception labels containing a total of 2800 instances (1400 gold and 1400 generated).

We asked native Italian speakers to assess the texts on a 1–5 Likert Scale asking whether they thought the text they were seeing had been produced by a human (1) or by a machine (5). Each text was assessed by seven different judges. The subjects for the task were laypeople recruited via the crowdsourcing platform Prolific[6]. We did not control for, and thus did not elicit, any demographic features. As a proxy for attention and quality control, we used completion time, and filtered out participants who took too little time to perform the task (we set an experimental threshold of at least 5 minutes for 70 assessments as a reliable minimum effort).[7]

Test Data To test the approach models we selected 1400 sentences, of which 700 are produced by human, 700 are automatically generated by GePpeTto, 700 are generated by the fine-tuned model and 700 by the reward-based model. We selected the human sentences in the same way as for the training data but picked the first 50 sentences with length 10, 15, 20, 25, 30, 35 and 40 tokens. We replicated the same approach for generating the training data for the three generation systems and we picked the first 50 sentences with length 10, 15, 20, 25, 30, 35 and 40 tokens for each system. Again for each sentence, we asked 5 users to evaluate if they perceive a sentence as human-produced or produced by an Artificial Intelligence on a Likert Scale from 1 to 5.

[6] https://www.prolific.co/

[7] Crowdworkers were compensated with a rate of 5.04 per estimated hour. In practice, tasks were completed in a shorter time than estimated, so the hourly rate was a bit higher.

7.6 Models

In our experiments, we use three models for automatic text generation, all based on the GPT-2 architecture. The basic model is `GePpeTto`, a GPT-2-based model for Italian released by [78]. The other two build on GePpeTto in two ways: first by fine-tuning it with perception-labelled data, and second by rewarding it in a reinforcement learning approach.

GePpeTto fine-tuned

GePpeTto is fine-tuned using the original settings of GePpeTto on the training portion of the perception-labelled data as described above, using the PyTorch implementation available on Huggingface Transformers wolf-etal-2020-transformers. We fine-tune GePpeTto with the optimiser is Adam diederik-kingma-2015 with an initial learning rate is 2e-5. The mini-batch size is set to 8. During fine-tuning, the early stopping with patience 5 is taken if the performance of the validation set does not improve.

The resulting model is expected to produce text which is recognised more frequently as human-produced than the original GePpeTto.

GePpeTto rewarded

In order to further encourage GePpeTto fine-tuned to generate more human-perceived texts, we introduce a confidence reward of 'style classifier' (SC). This is based on the confidence of a classifier based on UmBERTo[8] that is a Roberta [185] based Language Model pretrained on a large Italian corpus. The model has been fine-tuned on the perception-labelled data. In other words, the model is rewarded for generating more human-perceived text. SC's confidence is formulated as

$$R_{conf} = softmax_0(SC(y', \theta)) \tag{7.1}$$

where θ are the parameters of SC, which is fixed during fine-tuning GePpeTto. Formally, the confidence is used for policy learning that maximizes the expected reward $E[R]$ of the generated sequence, and the corresponding policy gradient is formulated as

$$\nabla_\phi E(R) = \nabla_\phi \sum_k (P(y_t^s | y_{1:t-1}^s; \phi) R_k \tag{7.2}$$

where ϕ are the parameters of GePpeTto, and R_k is the reward of the k_{th} sample sequence y^s, which is sampled from the model's distribution at each time step in decoding. Finally, the framework can be trained end-to-end by combining the policy gradient along with the cross-entropy loss of the base model.

BERT Regressor

In order to run automatic evaluation, we trained a regression model again based on UmBERTo. The model has been fine-tuned on the perception-labelled data.

[8] https://huggingface.co/Musixmatch/umberto-commoncrawl-cased-v1

7.7 Evaluation

We set up our evaluation in the following way. Each sentence was assigned the average score computed overall human judgements for that sentence. We then averaged all resulting scores over the seven length bins. This way, we obtain a single perception score per bin. Table 7.7 shows such averages for the four models: GePpeTto (gen), GePpeTto fine-tuned (gen1), GePpeTto rewarded (gen2) and the original human texts (*hum*). As a reminder, please note that the closer to 1, the more human-perceived the sentence. As a first observation, independently of length, we see that overall the human-produced texts are perceived as most human-like (score: 2.41). Regarding systems, the fine-tuned model performs better than both the basic model and the rewarded model. If we look deeper into the length aspect, we can make a series of interesting observations. First, at the shortest length, the automatic models (on average) and the human productions are evaluated in the same way, with gen1 being the most humanly-perceived model, even more so than the actual human texts. Second, the largest gap between humans and machines is observed in the longest sentences: if sentences are long and well-formed, they are human. Our best model overall (gen1) performs worse than the other models on longer sentences (35/40), where also the gap with human texts is the largest.

Table 7.8 shows the same evaluation framework using the scores that were produced by the BERT-regressor that is reused from the SC reward. The overall trend is comparable across the two evaluations and the final rank for the systems is the same. Two observations are necessary here. One is that the regressor tends to overestimate the performance of gen1 and gen2 due to the nature of their training. It also seems that the regressor is more fooled than humans with longer texts into thinking that these are human-produced. We also see that overall the regressor's scores are more compressed towards the middle than human judgements, which have a bit more variation.

Table 7.9 shows the correlation scores between human judgements and the regressor. The scores are calculated over each single data point. Both in terms of Pearson and RMSE, we can observe a reasonable correlation between the two judgements. Overall, comparing the automatic and the human-based evaluation shows that the latter might be preferable where available, but the former is a reliable strategy that can be always applied, even in absence of subjects.

	Length							
Tipo	10	15	20	25	30	35	40	AVG
gen	2.80	2.83	3.05	2.89	3.08	2.55	2.77	2.85
gen1	2.44	2.68	2.57	2.85	2.74	2.97	2.93	2.74
gen2	2.61	3.01	2.87	2.83	2.97	2.85	2.78	2.84
hum	2.59	2.45	2.38	2.37	2.48	2.39	2.18	2.41
avg	2.61	2.74	2.72	2.74	2.82	2.69	2.67	2.71

TABLE 7.7: Average scores for each system grouped by sentence length as assigned by humans on the test set.

	Length							
Tipo	10	15	20	25	30	35	40	AVG
gen	2.79	2.78	2.88	2.80	2.76	2.53	2.68	2.74
gen1	2.53	2.62	2.52	2.44	2.44	2.46	2.43	2.49
gen2	2.68	2.67	2.67	2.45	2.63	2.38	2.44	2.56
hum	2.74	2.70	2.38	2.55	2.51	2.20	2.16	2.47
avg	2.68	2.69	2.61	2.56	2.59	2.39	2.43	2.57

TABLE 7.8: Average scores for each system grouped by sentence length as assigned by the BERT based regressor on the test set

	MSE	**Pearson**	**Spearman**	**RMSE**
Test	0.56	0.54	0.51	0.75
Test hum	0.45	0.57	0.48	0.67
Test gen	0.53	0.57	0.56	0.73
Test gen1	0.55	0.55	0.51	0.74
Test gen2	0.7	0.47	0.43	0.84

TABLE 7.9: BERT Regressor scores for each systems

Further analysis on these data are ongoing but early results suggest that (i) human labelled data or a classifier can be used to improve the human-likeness of generated text (ii) a classifier can be used to approximate the human evaluation of human likeness. This result suggests that for what concerns human-likeness human evaluation is fundamental to improve and evaluate the capability of NLG systems, however, NLU techniques can be exploited at least to speed up the development and the evaluation of new models.

7.8 Conclusion

`GePpeTto` is the first GPT-2-based language model for Italian. Through both automatic and manual evaluation we assessed its quality on a variety of texts and in comparison to gold data as well as another statistical generation model. Results show that `GePpeTto` is able to produce a text which is much closer to human quality rather than to the text generated by the other generation model we have used. The linguistic analysis also highlights that `GePpeTto`'s production is quite similar to human production, though in a sort of bonsai version, since its sentences are on average shorter than the original texts, but with similar complexity.

The availability of `GePpeTto` opens up substantial possibilities. In the same way that GPT-2 is changing the approach to several NLP English tasks, we can expect `GePpeTto` to serve a similar purpose in Italian language processing. Moreover, results suggest that quite frequently human-produced sentences are perceived as automatically produced and vice-versa. Currently, we are working on a further study to model human-likeness: preliminary results showed that (i) a BERT based model is able to predict with good accuracy if a sentence will be perceived as human or automatically produced no matter if it is actually human or

automatically produced, (ii) the BERT based model can be used to make `GePpeTto` generate sentences more frequently perceived as human-produced.

Chapter 8

Conclusions

In this thesis machine learning methods to model stylistic aspects in Natural Language have been investigated. From one side we studied computational methodology to understand stylistic variation, from the other we examined how those techniques can be exploited to assess and improve the capabilities of NLG systems of reproducing such variations.

The first aspect considered is perceived as linguistic complexity. A method to model the human perception of sentence complexity relying on a new corpus of Italian and English sentences rated with human complexity judgments has been introduced. Moreover, the contribution of a wide set of linguistic features automatically extracted from these sentences in two experimental scenarios has been tested. The first one highlighted that we can reliably predict the degree of agreement between human annotators, independently from the assigned judgment of complexity. In the second experiment, we studied the correlation between linguistic features and complexity judgments. The presented corpus can be useful for different applications. From an NLP perspective, the corpus can be exploited to train systems able to predict people's perception of complexity. Moreover, the corpus can be exploited as well in Natural Language Generation tasks, going from text simplification to the automatic generation/evaluation of highly-engaging texts.

Also, we investigated deep learning methods for modelling several language variation aspects: sentiment analysis, hate speech detection, irony detection, author profiling. We firstly conducted a study on the effectiveness of multi-task learning approaches in sentiment polarity and irony classification. We presented a mixed single- and multi-task learning approach, that is able to improve the performance both in polarity and irony detection with respect to single-task and standard multi-task learning approaches. In particular, our approach led to substantial improvements on edge cases in which knowledge about the two tasks are needed to classify a tweet. This is particularly true when these cases are under-represented in the training data. An example is a case when a literal polarity of a tweet is inverted by irony.

Then we further tested our approach by participating in the ABSITA, GxG, HaSpeeDe and IronITA shared tasks of the EVALITA 2018 conference. By resorting to a system that used Support Vector Machines and DNN as learning algorithms, we achieved the best scores almost in every task. In addition, when DNN was used as a learning algorithm we applied the new multi-task learning approach and introduced a majority vote classification approach to further improve the overall accuracy of our system. The proposed system resulted in a very effective solution achieving the first position in almost all sub-tasks for each shared

task. Almost in all subtasks, neural approaches led to better performances than the SVM classifiers. However, for the GxG task, the accuracy scores are really low showing that for author profiling further works are needed. Another important aspect is that in cross-domain scenarios (HaSpeeDe and GxG) the systems obtained low scores indicating that the described approaches are not robust enough to deal with domain switching.

Finally, in 2020 we participated in the TAG-it task of EVALITA 2020. TAG-it is an Author Profiling task in which the goal is to provide a system capable of predicting the gender and the age of the authors of several blog posts and their topics. Our systems' performances showed that in the case in which the goal is to predict topic, age and gender dimensions at once, and in the case in which only the age must be predicted, the best classifier is the one developed using a Single-Task Learning approach and based on transformers. In the case in which the goal is the gender prediction only a Multi-task Learning approach combined with transformers have slightly better performances. These results prove that the proposed systems based on transformers are more effective than traditional machine learning techniques in the topic, age and gender classification achieving the state of the art for TAG-it shared task. Using deep pre-trained language models on this task Multi-Task Learning does not provide any relevant boost of performances. TAG-it could be seen as a continuation of the GxG task at EVALITA 2018. In the latter, teams were asked to predict gender within and across five different genres. We observe that results at TAG-it for gender prediction are higher than in GxG both within and cross-domain. This is might be ascribed to three main factors: (i) in this editions authors were represented by multiple texts, while in GxG, for some domains, evidence per author was minimal; (ii) texts in TAG-it are probably less noisy, at least in comparison to some of the GxG genres (e.g., tweets and YouTube comments); (iii) transformer based model (which were not widely available in 2018) provided a boost of performances.

In the second part of the thesis, we focused on variation in NLG we conducted several studies to model variation across the headlines of two Italian newspaper at the opposite of the political spectrum and we successfully applied this modelling method to evaluate the performances of several NLG systems we developed on two major aspects: (i) stylistic compliance in the respect of the target newspaper style, (ii) content preservation. In our first work, the quality of three different sequence-to-sequence models that generate headlines starting from an article was comparatively assessed through human judgement, which we contextually used to evaluate the original headlines as well. The best system is a pointer network model, with correctness judgements on par with the gold headlines. Evaluating the generated output on different levels, especially attractiveness, which typically characterises news headlines, uncovered an interesting aspect: gold headlines appear to be the most attractive to read the whole article, but are not considered the most suitable, on the contrary, they are judged as the most unsuitable of all. Therefore, when automatically generating headlines, just relying on content might never lead us to titles that are human-like and attractive enough for people to read the article. One aspect that we have not explicitly considered in our early experiments is that the headlines come from different newspapers (positioned at opposite ends of the political spectrum), and can carry newspaper-specific characteristics.

Then we experimented with using embeddings shifts as a tool to study how words are

used in two different Italian newspapers. We focused on a pre-selection of high-frequency words shared by the two newspapers, and on another set of words that were highlighted as potentially interesting through a newly proposed methodology that combines observed embeddings shifts and relative and absolute frequency. The most differently used words in the two newspapers are proper nouns of politically active individuals as well as places, and concepts that are highly debated on the political scene. Besides the present showcase, we believe this methodology can be more in general used to highlight which words might deserve deeper, dedicated analysis when studying meaning change. In another work on this topic, we trained a few pointer network models under different training settings that learnt to generate headlines according to a given newspaper's style, controlling for topic biases.

We also trained a few classifiers that are able to distinguish the source of a given headline with high accuracy. Using such classification models as evaluators we were able to verify that the generators we have trained are indeed style-aware. This was confirmed through an additional experiment that showed that if the headlines are generated by a model trained in a newspaper-agnostic fashion, the classifier is indeed not able to distinguish them. This whole evaluation procedure is done in a completely automated fashion. This is an advantage not only in terms of saving human effort but especially because our experiments suggest that humans cannot perform this task reliably enough. Turned out that coupling generation and classification appears to be a successful evaluation methodology which we believe can be applied more generally, especially in absence of reliable human judgement.

Finally, we addressed the issue of how to evaluate style transfer. We explicitly compared systems in terms of the extent to which they preserve content and their success at transferring style. The latter is known to be hard for humans to evaluate [85, 79]. Our aim was primarily to see to what extent different evaluation strategies based on purposely trained classifiers could distinguish between models, insofar as they perform better at either of these tasks and in different training scenarios. Our findings suggest that our proposed combination of classifiers focused on both content and style transfer can potentially help to distinguish models in terms of their strengths. Interestingly, a commonly used metric such as BLEU does not seem to be informative in our experiments, not even for the content preservation aspects. To the extent that stylistic distinctions remain hard for humans to evaluate in setups such as the one used here, a classification-based approach with consistency checks for content preservation is a promising way forward, especially to support development in a relatively cheap and effective way. Future work will have to determine how the various metrics we have used relate to each other (especially our classifiers and BLEU/ROUGE), and whether the human judgement can be successfully brought back, and in the case in what form, at some stage of the evaluation process. The task we framed has been proposed at EVALITA 2020 as the first NLG shared task ever presented in the EVALITA campaigns.

In the last part of the thesis, we introduced `GePpeTto` : the first GPT-2-based language model for Italian. Through both automatic and manual evaluation we assessed its quality on a variety of texts and in comparison to gold data as well as another statistical generation model. Results show that `GePpeTto` is able to produce texts which are much closer to human quality rather than to the text generated by the other generation model we have used.

The linguistic analysis also highlights that GePpeTto's production is quite similar to human production, though in a sort of bonsai version since its sentences are on average shorter than the original texts, but with similar complexity. The availability of GePpeTto opens up substantial possibilities. In the same way that GPT-2 is changing the approach to several NLP English tasks, we can expect GePpeTto to serve a similar purpose in Italian language processing. Moreover, results suggest that quite frequently human-produced sentences are perceived as automatically produced. A further work about human-likeness of GePpeTto produced texts is ongoing but early results suggest that (i) human labelled data or a classifier can be used to improve the human-likeness of generated text (ii) a classifier can be used to approximate the human evaluation of human-likeness of generated texts. This result suggests that for what concerns human-likeness humans contribution is helpful to improve and evaluate the capability of NLG systems, however, NLU techniques can be exploited at least to speed up the development and the evaluation of new models.

More in general in this thesis we highlighted the strengths and weaknesses of human-made variation assessment both in automatic and human-produced texts, and we highlighted some scenarios in which machine learning methods can be used on completion to assess variational aspects. This contribution provides the research community with a clear indication of how to perform the evaluation of several aspects of NLG systems.

All the work done in this thesis has been done for the Italian language, while doing it we produced several models, data and resources for the Italian NLP community. Working on the Italian language was not easy since it is a low-resource language compared to other languages for which a wide range of datasets and pre-trained models are available. However, by doing it we obtained two important results: (i) we produced data and models that could stimulate the Italian NLP community in producing research contributions on variational aspects modelling and in NLG, (ii) we produced research results that can be replicated for other low-resource languages.

Appendix A

Publications and Awards

During the PhD project the following works has been published:

- Brunato, Dominique; De Mattei, Lorenzo; Dell'Orletta, Felice; Iavarone, Benedetta; Venturi, Giulia; Is this Sentence Difficult? Do you Agree? Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing 2690-2699. 2018.
- De Mattei, Lorenzo; Cimino, Andrea; Dell'Orletta, Felice; Multi-Task Learning in Deep Neural Network for Sentiment Polarity and Irony classification. NL4AI@ AI*IA 76-82. 2018.
- Cimino, Andrea; De Mattei, Lorenzo; Dell'Orletta, Felice; Multi-task learning in deep neural networks at Evalita 2018. Proceedings of the Evaluation Campaign of Natural Language Processing and Speech tools for Italian. 86-952018.
- Cafagna, Michele; De Mattei, Lorenzo; Nissim, Malvina; Embeddings Shifts as Proxies for Different Word Use in Italian Newspapers. CLiC-it, 2019.
- Cafagna, Michele; De Mattei, Lorenzo; Bacciu, Davide; Nissim, Malvina; Suitable Doesn't Mean Attractive. Human-Based Evaluation of Automatically Generated Headlines. CLiC-it, 2019.
- De Mattei, Lorenzo; Cafagna, Michele; Dell'Orletta, Felice; Nissim, Malvina; Guerini, Marco; Geppetto carves italian into a language model. CLiC-it, 2020.
- van der Goot, Rob; Ramponi, Alan; Caselli, Tommaso; Cafagna, Michele; De Mattei, Lorenzo; Norm It! Lexical Normalization for Italian and Its Downstream Effects for Dependency Parsing. Proceedings of The 12th Language Resources and Evaluation Conference. 6272-6278. 2020.
- De Mattei, Lorenzo; Cafagna, Michele; Dell'Orletta, Felice; Nissim, Malvina; Invisible to people but not to machines: Evaluation of style-aware HeadlineGeneration in absence of reliable human judgment. Proceedings of The 12th Language Resources and Evaluation Conference. 6709-6717. 2020.
- De Mattei, Lorenzo; De Martino, Graziella; Iovine, Andrea; Miaschi, Alessio; Polignano, Marco; Rambelli, Giulia; ATE ABSITA @ EVALITA2020: Overview of the aspect term extraction and aspect-based sentiment analysis task. Proceedings of the

7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020), Online. CEUR.

- De Mattei, Lorenzo; Cafagana, Michele; Dell'Orletta, Felice; Nissim, Malvina; Gatt, Albert; CHANGE-IT@ EVALITA 2020: Change Headlines, Adapt News, GEnerate. Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020), Online. CEUR. org2020
- Occhipinti, Daniela; Tesei, Andrea; Iacono, Maria; Aliprandi, Carlo; De Mattei, Lorenzo; ItaliaNLP@ TAG-IT: UmBERTo for Author Profiling at TAG-it 2020. Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020), Online. CEUR.
- De Mattei, Lorenzo; Cafagna, Michele; Lai, Huiyuan; Dell'Orletta, Felice; Nissim, Malvina; Gatt, Albert; On the interaction of automatic evaluation and task framing in headline style transfer. Workshop on Evaluating NLG Evaluation at INLG 2020. Online from Dublin, Ireland, 18 December 2020.
- Cafagna, Michele; De Mattei, Lorenzo; Nissim, Malvina. "Embeddings-based detection of word use variation in Italian newspapers", IJCoL, 6-2

The candidate received the following awards:

- Best System Award Across Tasks at EVALITA 2018. https://www.evalita.it/2018/best-system
- Special Mention at CLiC-it 2019 for the paper: "Embeddings Shifts as Proxies for Different Word Use in Italian Newspapers"

Bibliography

[1] Eneko Agirre et al. “SemEval-2016 Task 2: Interpretable Semantic Textual Similarity”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 512–524. DOI: 10.18653/v1/S16-1082. URL: https://www.aclweb.org/anthology/S16-1082.

[2] Jacopo Amidei, Paul Piwek, and Alistair Willis. “The use of rating and Likert scales in Natural Language Generation human evaluation tasks: A review and some recommendations”. In: *Proceedings of the 12th International Conference on Natural Language Generation*. Tokyo, Japan: Association for Computational Linguistics, Oct. 2019, pp. 397–402. DOI: 10.18653/v1/W19-8648. URL: https://www.aclweb.org/anthology/W19-8648.

[3] Peter Anderson et al. “SPICE: Semantic Propositional Image Caption Evaluation”. In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 382–398. ISBN: 978-3-319-46454-1.

[4] Nabiha Asghar et al. “Affective Neural Response Generation”. In: *European Conference on Information Retrieval*. Springer. 2018, pp. 154–166.

[5] Giuseppe Attardi. *Wikiextractor*. http://attardi.github.io/wikiextractor. 2012.

[6] Giuseppe Attardi et al. “Convolutional Neural Networks for Sentiment Analysis on Italian Tweets.” In: *CLiC-it/EVALITA*. 2016.

[7] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer normalization”. In: *arXiv preprint arXiv:1607.06450* (2016).

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1409.0473.

[9] Hadi Banaee, Mobyen Uddin Ahmed, and Amy Loutfi. “A framework for automatic text generation of trends in physiological time series data”. In: *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. IEEE. 2013, pp. 3876–3881.

[10] Colin Bannard and Chris Callison-Burch. "Paraphrasing with bilingual parallel corpora". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics. 2005, pp. 597–604.

[11] Francesco Barbieri et al. "Overview of the evalita 2016 sentiment polarity classification task". In: *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*. 2016.

[12] Ellen Bard and Dan Robertson. "Magnitude Estimation of Linguistic Acceptability". In: *Language* 72 (Mar. 1996). DOI: 10.2307/416793.

[13] Kobus Barnard. "Computational methods for integrating vision and language". In: *Synthesis Lectures on Computer Vision* 6.1 (2016), pp. 1–227.

[14] Marco Baroni and Silvia Bernardini. "A new approach to the study of translationese: Machine-learning the difference between original and translated text". In: *Literary and Linguistic Computing* 21.3 (2005), pp. 259–274.

[15] Marco Baroni et al. "The WaCky wide web: a collection of very large linguistically processed web-crawled corpora". In: *Language resources and evaluation* 43.3 (2009), pp. 209–226.

[16] Alberto Bartoli et al. "Your paper has been accepted, rejected, or whatever: Automatic generation of scientific paper reviews". In: *International Conference on Availability, Reliability, and Security*. Springer. 2016, pp. 19–28.

[17] Regina Barzilay and Lillian Lee. "Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment". In: *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*. 2003, pp. 16–23. URL: https://www.aclweb.org/anthology/N03-1003.

[18] Pierpaolo Basile et al. "O verview of the EVALITA 2018 Aspect-based Sentiment Analysis task (ABSITA)". In: *EVALITA Evaluation of NLP and Speech Tools for Italian* 12 (2018), p. 10.

[19] Valerio Basile and Malvina Nissim. "Sentiment analysis on Italian tweets". In: *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Atlanta, 2013, pp. 100–107.

[20] Valerio Basile et al. "EVALITA 2020: Overview of the 7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian". In: *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*. Ed. by Valerio Basile et al. Online: CEUR.org, 2020.

[21] Susana Bautista and Horacio Saggion. "Can Numerical Expressions Be Simpler? Implementation and Demostration of a Numerical Simplification System for Spanish". In: *Proceedings of LREC, the 9th International Conference on Language Resources and Evaluation*. Reykjavik, Iceland, 2014.

[22] Iz Beltagy, Matthew E Peters, and Arman Cohan. "Longformer: The long-document transformer". In: *arXiv preprint arXiv:2004.05150* (2020).

[23] Anja Belz and Eric Kow. "Comparing rating scales and preference judgements in language evaluation". In: *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics. 2010, pp. 7–15.

[24] Anja Belz and Ehud Reiter. "Comparing Automatic and Human Evaluation of NLG Systems". In: *11th Conference of the European Chapter of the Association for Computational Linguistics*. Trento, Italy: Association for Computational Linguistics, Apr. 2006. URL: https://www.aclweb.org/anthology/E06-1040.

[25] Emily M. Bender et al. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 610–623. ISBN: 9781450383097. DOI: 10.1145/3442188.3445922. URL: https://doi.org/10.1145/3442188.3445922.

[26] Yoshua Bengio et al. "A neural probabilistic language model". In: *Journal of machine learning research* 3.Feb (2003), pp. 1137–1155.

[27] Gaetano Berruto. *Fondamenti di sociolinguistica*. Laterza, 1995.

[28] Douglas Biber. *Variation across speech and writing*. Cambridge University Press, 1991.

[29] Alan W Black et al. "Spoken Dialog Challenge 2010: Comparison of Live and Control Test Results". In: *Proceedings of the SIGDIAL 2011 Conference*. Portland, Oregon: Association for Computational Linguistics, June 2011, pp. 2–7. URL: https://www.aclweb.org/anthology/W11-2002.

[30] Tolga Bolukbasi et al. "Quantifying and reducing stereotypes in word embeddings". In: *arXiv preprint arXiv:1606.06121* (2016).

[31] Cristina Bosco et al. "Overview of the EVALITA 2018 HaSpeeDe Hate Speech Detection (HaSpeeDe) Task". In: *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA18), Turin, Italy, December. CEUR. org* (2018).

[32] Samuel R. Bowman et al. "A large annotated corpus for learning natural language inference". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 632–642. DOI: 10.18653/v1/D15-1075. URL: https://www.aclweb.org/anthology/D15-1075.

[33] Jonathan C Brown, Gwen A Frishkoff, and Maxine Eskenazi. "Automatic question generation for vocabulary assessment". In: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2005, pp. 819–826.

[34] Peter F Brown et al. “Class-based n-gram models of natural language”. In: *Computational linguistics* 18.4 (1992), pp. 467–480.

[35] Tom B Brown et al. “Language models are few-shot learners”. In: *arXiv preprint arXiv:2005.14165* (2020).

[36] D. Brunato et al. “PaCCSS–IT: A Parallel Corpus of Complex–Simple Sentences for Automatic Text Simplification”. In: *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*. Austin, Texas, USA, 2016, pp. 10–18.

[37] Dominique Brunato et al. “Is this Sentence Difficult? Do you Agree?” In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, pp. 2690–2699.

[38] Dominique Brunato et al. “Profiling-UD: a Tool for Linguistic Profiling of Texts”. English. In: *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 7145–7151. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.883.

[39] Michele Cafagna, Lorenzo De Mattei, and Malvina Nissim. “Embeddings Shifts as Proxies for Different Word Use in Italian Newspapers.” In: *CLiC-it*. 2019.

[40] Michele Cafagna et al. “Suitable Doesn’t Mean Attractive. Human-Based Evaluation of Automatically Generated Headlines”. In: *CLiC-it*. 2019.

[41] Aoife Cahill and Martin Forst. “Human Evaluation of a German Surface Realisation Ranker”. In: *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. Athens, Greece: Association for Computational Linguistics, Mar. 2009, pp. 112–120. URL: https://www.aclweb.org/anthology/E09-1014.

[42] Aoife Cahill and Martin Forst. “Human evaluation of a German surface realisation ranker”. In: *Empirical methods in natural language generation*. Springer, 2009, pp. 201–221.

[43] Chris Callison-Burch et al. “(Meta-) Evaluation of Machine Translation”. In: *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 136–158. URL: https://www.aclweb.org/anthology/W07-0718.

[44] Tommaso Caselli et al. “Evalita 2018: Overview on the 6th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian”. In: Jan. 2018, pp. 3–8. ISBN: 9788831978422. DOI: 10.4000/books.aaccademia.4437.

[45] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. *Evaluation of Text Generation: A Survey*. 2020. arXiv: 2006.14799 [cs.CL].

[46] Asli Celikyilmaz et al. "Deep Communicating Agents for Abstractive Summarization". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1662–1675. DOI: 10.18653/v1/N18-1150. URL: https://www.aclweb.org/anthology/N18-1150.

[47] Eugene Charniak. "A maximum-entropy-inspired parser". In: *1st Meeting of the North American Chapter of the Association for Computational Linguistics*. 2000.

[48] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[49] David L Chen and Raymond J Mooney. "Learning to sportscast: a test of grounded language acquisition". In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, pp. 128–135.

[50] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. URL: https://www.aclweb.org/anthology/D14-1179.

[51] Kyunghyun Cho et al. "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches". In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111. DOI: 10.3115/v1/W14-4012. URL: https://www.aclweb.org/anthology/W14-4012.

[52] Krzysztof Choromanski et al. "Rethinking attention with performers". In: *arXiv preprint arXiv:2009.14794* (2020).

[53] Jan K Chorowski et al. "Attention-based models for speech recognition". In: *Advances in neural information processing systems*. 2015, pp. 577–585.

[54] Alessandra Teresa Cignarella et al. "Overview of the evalita 2018 task on irony detection in italian tweets (ironita)". In: *Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2018)*. Vol. 2263. CEUR-WS. 2018, pp. 1–6.

[55] Andrea Cimino and Felice Dell'Orletta. "Building the state-of-the-art in POS tagging of Italian Tweets." In: *CLiC-it/EVALITA*. 2016.

[56] Andrea Cimino and Felice Dell'Orletta. "Tandem LSTM-SVM approach for sentiment analysis". In: *of the Final Workshop 7 December 2016, Naples*. 2016, p. 172.

[57] Andrea Cimino, Dell'Orletta Felice, and Nissim Malvina. "TAG-it@EVALITA2020: Overview of the Topic, Age, and Gender prediction task for Italian". In: *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*. Ed. by Valerio Basile et al. Online: CEUR.org, 2020.

[58] Andrea Cimino, Lorenzo De Mattei, and Felice Dell'Orletta. "Multi-task Learning in Deep Neural Networks at EVALITA 2018". In: *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy, December 12-13, 2018*. Vol. 2263. CEUR Workshop Proceedings. CEUR-WS.org, 2018. URL: http://ceur-ws.org/Vol-2263/paper013.pdf.

[59] Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. "Sentence Mover's Similarity: Automatic Evaluation for Multi-Sentence Texts". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 2748–2760. DOI: 10.18653/v1/P19-1264. URL: https://www.aclweb.org/anthology/P19-1264.

[60] James Clarke and Mirella Lapata. "Discourse constraints for document compression". In: *Computational Linguistics* 36.3 (2010), pp. 411–441.

[61] O. De Clercq et al. "Using the crowd for readability prediction". In: *Natural Language Engineering* (2013), pp. 1–33.

[62] Christer Clerwall. "Enter the robot journalist: Users' perceptions of automated content". In: *Journalism Practice* 8.5 (2014), pp. 519–531.

[63] Jacob Cohen. "A Coefficient of Agreement for Nominal Scales". In: *Educational and Psychological Measurement* 20.1 (1960), pp. 37–46. DOI: 10.1177/001316446002000104. eprint: https://doi.org/10.1177/001316446002000104. URL: https://doi.org/10.1177/001316446002000104.

[64] Michael Collins. "Three generative, lexicalised models for statistical parsing". In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. 1997, pp. 16–23.

[65] Kevyn Collins-Thompson. "Computational assessment of text readability: A survey of current and future research". In: *Recent Advances in Automatic Readability Assessment and Text Simplification. Special issue of International Journal of Applied Linguistics* 165.2 (2015), pp. 97–135.

[66] Ronan Collobert and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 160–167.

[67] Ronan Collobert et al. "Natural language processing (almost) from scratch". In: *Journal of machine learning research* 12.ARTICLE (2011), pp. 2493–2537.

[68] Carlos A Colmenares et al. "HEADS: Headline generation as sequence prediction using an abstract feature-rich space". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2015, pp. 133–142.

[69] Alexis Conneau et al. "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (2017). DOI: 10.18653/v1/d17-1070. URL: http://dx.doi.org/10.18653/v1/D17-1070.

[70] Walter Daelemans. "Explanation in computational stylometry". In: *International conference on intelligent text processing and computational linguistics*. Springer. 2013, pp. 451–462.

[71] Ido Dagan, Oren Glickman, and Bernardo Magnini. "The PASCAL Recognising Textual Entailment Challenge". In: *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*. Ed. by Joaquin Quiñonero-Candela et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 177–190. ISBN: 978-3-540-33428-6.

[72] Daniel Dahlmeier, Chang Liu, and Hwee Tou Ng. "TESLA at WMT 2011: Translation Evaluation and Tunable Metric". In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, July 2011, pp. 78–84. URL: https://www.aclweb.org/anthology/W11-2106.

[73] Robert Dale, Ilya Anisimoff, and George Narroway. "HOO 2012: A report on the preposition and determiner error correction shared task". In: *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics. 2012, pp. 54–62.

[74] Sumanth Dathathri et al. *Plug and Play Language Models: A Simple Approach to Controlled Text Generation*. 2020. arXiv: 1912.02164 [cs.CL].

[75] L De Mattei et al. "On the interaction of automatic evaluation and task framing in headline style transfer". In: *Proceedings of the 1st Workshop on Evaluating NLG Evaluation (EvalNLGEval'20)*. Dublin, Ireland: Association for Computational Linguistics, 2020. URL: https://evalnlg-workshop.github.io/papers/EvalNLGEval_2020_paper_8.pdf.

[76] Lorenzo De Mattei and Andrea Cimino. "Multi-Task Learning in Deep Neural Network for Sentiment Polarity and Irony classification." In: *Proceedings of the 2nd Workshop on Natural Language for Artificial Intelligence*. Nov. 2018.

[77] Lorenzo De Mattei et al. "CHANGE-IT @ EVALITA 2020: Change Headlines, Adapt News, GEnerate". In: *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*. Ed. by Valerio Basile et al. Online: CEUR.org, 2020.

[78] Lorenzo De Mattei et al. "GePpeTto Carves Italian into a Language Model". In: *Proceedings of the Seventh Italian Conference on Computational Linguistics, CLiC-it 2020, Bologna, Italy, March 1-3, 2021*. Ed. by Johanna Monti, Felice Dell'Orletta, and Fabio Tamburini. Vol. 2769. CEUR Workshop Proceedings. CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2769/paper%5C_46.pdf.

[79] Lorenzo De Mattei et al. "Invisible to People but not to Machines: Evaluation of Style-aware HeadlineGeneration in Absence of Reliable Human Judgment". English. In: *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 6709–6717. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.828.

[80] Marco Del Tredici, Malvina Nissim, and Andrea Zaninello. "Tracing metaphors in time through self-distance in vector spaces". In: *Proceedings of the Third Italian Conference on Computational Linguistics (CLiC-it 2016)*. 2016.

[81] Felice Dell'Orletta and Malvina Nissim. *Overview of the EVALITA 2018 cross-genre gender prediction (gxg) task*. 2018.

[82] Vera Demberg and Frank Keller. "A Computational Model of Prediction in Human Parsing: Unifying Locality and Surprisal Effects". In: *Proceedings of CogSci 2009, the 31st Annual Conference of the Cognitive Science Society*. 2009, pp. 1888–1893.

[83] Jan Deriu et al. "Survey on evaluation methods for dialogue systems". In: *Artificial Intelligence Review* (June 2020). ISSN: 1573-7462. DOI: 10.1007/s10462-020-09866-x. URL: http://dx.doi.org/10.1007/s10462-020-09866-x.

[84] Jan Milan Deriu and Mark Cieliebak. "Sentiment analysis using convolutional neural networks with multi-task training and distant supervision on italian tweets". In: *Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Napoli, Italy, December 5-7, 2016*. Italian Journal of Computational Linguistics. 2016.

[85] Nina Dethlefs et al. "Cluster-based Prediction of User Ratings for Stylistic Surface Realisation". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, Apr. 2014, pp. 702–711. DOI: 10.3115/v1/E14-1074. URL: https://www.aclweb.org/anthology/E14-1074.

[86] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://www.aclweb.org/anthology/N19-1423.

[87] Djellel Difallah, Elena Filatova, and Panos Ipeirotis. "Demographics and Dynamics of Mechanical Turk Workers". In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. WSDM '18. Marina Del Rey, CA, USA: Association for Computing Machinery, 2018, pp. 135–143. ISBN: 9781450355810. DOI: 10.1145/3159652.3159661. URL: https://doi.org/10.1145/3159652.3159661.

[88] William B. Dolan and Chris Brockett. "Automatically Constructing a Corpus of Sentential Paraphrases". In: *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*. 2005. URL: https://www.aclweb.org/anthology/I05-5002.

[89] Daxiang Dong et al. "Multi-task learning for multiple language translation". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Vol. 1. 2015, pp. 1723–1732.

[90] Li Dong et al. "Learning to generate product reviews from attributes". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Vol. 1. 2017, pp. 623–632.

[91] Jeffrey L Elman. "Finding structure in time". In: *Cognitive science* 14.2 (1990), pp. 179–211.

[92] Matan Eyal, Tal Baumel, and Michael Elhadad. "Question Answering as an Automatic Evaluation Metric for News Article Summarization". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 3938–3948. DOI: 10.18653/v1/N19-1395. URL: https://www.aclweb.org/anthology/N19-1395.

[93] Christiane Fellbaum. "WordNet". In: *The encyclopedia of applied linguistics* (2012).

[94] Thiago Castro Ferreira et al. "Linguistic realisation as machine translation: Comparing different MT models for AMR-to-text generation". In: *Proceedings of the 10th International Conference on Natural Language Generation*. 2017, pp. 1–10.

[95] Jessica Ficler and Yoav Goldberg. "Controlling linguistic style aspects in neural language generation". In: *arXiv preprint arXiv:1707.02633* (2017).

[96] Charles J Fillmore, Josef Ruppenhofer, and Collin F Baker. "Framenet and representing the link between semantic and syntactic relations". In: *Frontiers in linguistics* 1 (2004), pp. 19–59.

[97] John R Firth. "A synopsis of linguistic theory, 1930-1955". In: *Studies in linguistic analysis* (1957).

[98] John Rupert Firth. "The Technique of Semantics." In: *Transactions of the philological society* 34.1 (1935), pp. 36–73.

[99] Joseph L Fleiss. "Measuring nominal scale agreement among many raters." In: *Psychological bulletin* 76.5 (1971), p. 378.

[100] Lucie Flekova, Daniel Preoţiuc-Pietro, and Lyle Ungar. "Exploring Stylistic Variation with Age and Income on Twitter". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 313–319. DOI: 10.18653/v1/P16-2051. URL: https://www.aclweb.org/anthology/P16-2051.

[101] Lin Frazier. "Syntactic complexity". In: *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives* Cambridge: Cambridge University Press (1985), pp. 129–189.

[102] Zhenxin Fu et al. "Style transfer in text: Exploration and evaluation". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[103] Philip Gage. "A new algorithm for data compression". In: *C Users Journal* 12.2 (1994), pp. 23–38.

[104] Yarin Gal and Zoubin Ghahramani. "A theoretically grounded application of dropout in recurrent neural networks". In: *Advances in neural information processing systems* 29 (2016), pp. 1019–1027.

[105] Albert Gatt and Emiel Krahmer. "Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation". In: *Journal of Artificial Intelligence Research* 61 (2018), pp. 65–170.

[106] Albert Gatt et al. "From data to text in the neonatal intensive care unit: Using NLG technology for decision support and information management". In: *Ai Communications* 22.3 (2009), pp. 153–186.

[107] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. "Gltr: Statistical detection and visualization of generated text". In: *arXiv preprint arXiv:1906.04043* (2019).

[108] Sayan Ghosh et al. "Affect-lm: A neural language model for customizable affective text generation". In: *arXiv preprint arXiv:1704.06851* (2017).

[109] Edward Gibson. "Linguistic complexity: Locality of syntactic dependencies". In: *Cognition* 24.11 (1998), pp. 1–76.

[110] Edward Gibson. "The dependency Locality Theory: A distance–based theory of linguistic complexity". In: *W.O.A. Marants and Y. Miyashita (Eds.), Image, Language and Brain* Cambridge, MA: MIT Press (2000), pp. 95–126.

[111] Edward Gibson and Neal J. Pearlmutter. "A corpus-based analysis of psycholinguistic constraints on prepositional-phrase attachment". In: *Perspectives on sentence processing* (1994), pp. 181–198.

[112] Timothy R. Gibson. "Towards a discourse theory of abstracts and abstracting". In: *Monographs in Systemic Linguistics* Nottingham: Department of English Studies, University of Nottingham (1993).

[113] Daniel Gildea and David Temperley. "Do Grammars Minimize Dependency Length?" In: *Cognitive Science* 34.2 (2010), pp. 286–310.

[114] Eli Goldberg, Norbert Driedger, and Richard I Kittredge. "Using natural-language processing to produce weather forecasts". In: *IEEE Intelligent Systems* 2 (1994), pp. 45–53.

[115] Yoav Goldberg. "Neural network methods for natural language processing". In: *Synthesis Lectures on Human Language Technologies* 10.1 (2017), pp. 1–309.

[116] Hongyu Gong et al. "Reinforcement Learning Based Text Style Transfer without Parallel Training Corpus". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019, pp. 3168–3180.

[117] Li Gong, Josep Crego, and Jean Senellart. "Enhanced Transformer Model for Data-to-Text Generation". In: *Proceedings of the 3rd Workshop on Neural Generation and Translation*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 148–156. DOI: 10.18653/v1/D19-5615. URL: https://www.aclweb.org/anthology/D19-5615.

[118] Rob van der Goot et al. "Bleaching Text: Abstract Features for Cross-lingual Gender Prediction". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 383–389. DOI: 10.18653/v1/P18-2061. URL: https://www.aclweb.org/anthology/P18-2061.

[119] Peter C. Gordon, Randall Hendrick, and Marcus Johnson. "Memory interference during language processing". In: *Journal of Experimental Psychology: Learning, Memory and Cognition* 27.6 (2001), pp. 1411–1423.

[120] GPT-3. "A robot wrote this entire article. Are you scared yet, human?" In: *The Guardian* (Sept. 8, 2020). URL: https://www.theguardian.com/commentisfree/2020/sep/08/robot-wrote-this-article-gpt-3 (visited on 09/23/2020).

[121] Yvette Graham and Timothy Baldwin. "Testing for Significance of Increased Correlation with Human Judgment". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 172–176. DOI: 10.3115/v1/D14-1020. URL: https://www.aclweb.org/anthology/D14-1020.

[122] Alex Graves and Jürgen Schmidhuber. "Framewise phoneme classification with bidirectional LSTM and other neural network architectures". In: *Neural Networks* 18.5-6 (2005), pp. 602–610.

[123] Riccardo Guidotti et al. "A Survey of Methods for Explaining Black Box Models". In: *ACM Comput. Surv.* 51.5 (Aug. 2018). ISSN: 0360-0300. DOI: 10.1145/3236009. URL: https://doi.org/10.1145/3236009.

[124] Kristina Gulordava and Paola Merlo. "Diachronic Trends in Word Order Freedom and Dependency Length in Dependency-Annotated Corpora of Latin and Ancient Greek". In: *Proceedings of Depling 2015, the Third International Conference on Dependency Linguistics*. 2015.

[125] John Hale. "A Probabilistic Earley Parser as a Psycholinguistic Model". In: *Proceedings of the NAACL*. 2001, pp. 159–166.

[126] William L Hamilton, Jure Leskovec, and Dan Jurafsky. "Cultural shift or linguistic drift? comparing two computational measures of semantic change". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*. Vol. 2016. NIH Public Access. 2016, p. 2116.

[127] William L Hamilton, Jure Leskovec, and Dan Jurafsky. "Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 1489–1501.

[128] Aaron L.-F Han et al. "Unsupervised Quality Estimation Model for English to German Translation and Its Application in Extensive Supervised Evaluation". In: *TheScientificWorldJOURNAL* (Dec. 2013). DOI: 10.1155/2014/760301.

[129] Zellig S Harris. "Distributional structure". In: *Word* 10.2-3 (1954), pp. 146–162.

[130] Tatsunori Hashimoto, Hugh Zhang, and Percy Liang. "Unifying Human and Statistical Evaluation for Natural Language Generation". In: *Proceedings of the 2019 Conference of the North* (2019). DOI: 10.18653/v1/n19-1169. URL: http://dx.doi.org/10.18653/v1/N19-1169.

[131] Helen Hastie and Anja Belz. "A Comparative Evaluation Methodology for NLG in Interactive Systems". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014, pp. 4004–4011. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/1147_Paper.pdf.

[132] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[133] Jonathan Herzig et al. "Neural response generation for customer service based on personality traits". In: *Proceedings of the 10th International Conference on Natural Language Generation*. 2017, pp. 252–256.

[134] Martin Heusel et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6629–6640. ISBN: 9781510860964.

[135] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://doi.org/10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[136] Ari Holtzman et al. *The Curious Case of Neural Text Degeneration*. 2019. arXiv: 1904.09751 [cs.CL].

[137] Jeremy Howard and Sebastian Ruder. "Universal Language Model Fine-tuning for Text Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 328–339. DOI: 10.18653/v1/P18-1031. URL: https://www.aclweb.org/anthology/P18-1031.

[138] Zhiting Hu et al. "Controllable text generation. arXiv preprint". In: *arXiv preprint arXiv:1703.00955* 7 (2017).

[139] Po-Sen Huang et al. "Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data". In: *Proceedings of the 22nd ACM International Conference on Information amp; Knowledge Management*. CIKM '13. San Francisco, California, USA: Association for Computing Machinery, 2013, pp. 2333–2338. ISBN: 9781450322638. DOI: 10.1145/2505515.2505665. URL: https://doi.org/10.1145/2505515.2505665.

[140] Kellogg W. Hunt. "Recent Measures in Syntactic Development". In: *Elementary English* 43.7 (1966), pp. 732–739.

[141] Dirk Hüske-Kraus. "Text generation in clinical medicine–a review". In: *Methods of information in medicine* 42.01 (2003), pp. 51–60.

[142] William John Hutchins and Harold L Somers. *An introduction to machine translation*. Vol. 362. Academic Press London, 1992.

[143] Panos Ipeirotis, Foster Provost, and Jing Wang. "Quality Management on Amazon Mechanical Turk". In: *In:Proceedings of the ACM SIGKDD Workshop on Human Computation* (Oct. 2010). DOI: 10.1145/1837885.1837906.

[144] Aylin Caliskan Islam, Joanna J Bryson, and Arvind Narayanan. "Semantics derived automatically from language corpora necessarily contain human biases". In: *CoRR, abs/1608.07187* (2016).

[145] Vineet John et al. "Disentangled representation learning for text style transfer". In: *arXiv preprint arXiv:1808.04339* (2018).

[146] Karen Sparck Jones. "Natural language processing: a historical review". In: *Current issues in computational linguistics: in honour of Don Walker*. Springer, 1994, pp. 3–16.

[147] Karen Sparck Jones and Julia R. Galliers. *Evaluating Natural Language Processing Systems: An Analysis and Review*. Berlin, Heidelberg: Springer-Verlag, 1996. ISBN: 3540613099.

[148] Nal Kalchbrenner and Phil Blunsom. "Recurrent continuous translation models". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 2013, pp. 1700–1709.

[149] Hassan Kané et al. "Towards Neural Similarity Evaluator". In: *Workshop on Document Intelligence at NeurIPS 2019*. 2019. URL: https://openreview.net/forum?id=S1xkQac9LB.

[150] David Kauchak and Regina Barzilay. "Paraphrasing for automatic evaluation". In: *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics. 2006, pp. 455–462.

[151] Nitish Shirish Keskar et al. "Ctrl: A conditional transformer language model for controllable generation". In: *arXiv preprint arXiv:1909.05858* (2019).

[152] Mert Kilickaya et al. "Re-evaluating Automatic Metrics for Image Captioning". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 199–209. URL: https://www.aclweb.org/anthology/E17-1019.

[153] Yoon Kim et al. "Temporal Analysis of Language through Neural Language Models". In: *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. Baltimore, MD, USA: Association for Computational Linguistics, June 2014, pp. 61–65. URL: http://www.aclweb.org/anthology/W/W14/W14-2517.

[154] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[155] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. URL: http://arxiv.org/abs/1312.6114.

[156] Svetlana Kiritchenko and Saif M. Mohammad. "Capturing Reliable Fine-Grained Sentiment Associations by Crowdsourcing and Best–Worst Scaling". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 811–817. DOI: 10.18653/v1/N16-1095. URL: https://www.aclweb.org/anthology/N16-1095.

[157] Ryan Kiros et al. *Skip-Thought Vectors*. 2015. arXiv: 1506.06726 [cs.CL].

[158] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. "Reformer: The efficient transformer". In: *arXiv preprint arXiv:2001.04451* (2020).

[159] Guillaume Klein et al. "OpenNMT: Open-Source Toolkit for Neural Machine Translation". In: *Proc. ACL*. 2017. DOI: 10.18653/v1/P17-4012. URL: https://doi.org/10.18653/v1/P17-4012.

[160] Rik Koncel-Kedziorski et al. "Text Generation from Knowledge Graphs with Graph Transformers". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 2284–2293. DOI: 10.18653/v1/N19-1238. URL: https://www.aclweb.org/anthology/N19-1238.

[161] Moshe Koppel, Shlomo Argamon, and Anat Rachel Shimoni. "Automatically categorizing written texts by author gender". In: *Literary and linguistic computing* 17.4 (2002), pp. 401–412.

[162] Ben Krause et al. "GeDi: Generative Discriminator Guided Sequence Generation". In: *ArXiv* abs/2009.06367 (2020).

[163] Nikolaus Kriegeskorte and Rogier A Kievit. "Representational geometry: integrating cognition, computation, and the brain". In: *Trends in cognitive sciences* 17.8 (2013), pp. 401–412.

[164] Klaus Krippendorff. "Estimating the Reliability, Systematic Error and Random Error of Interval Data". In: *Educational and Psychological Measurement* 30.1 (1970), pp. 61–70. DOI: 10.1177/001316447003000105. eprint: https://doi.org/10.1177/001316447003000105. URL: https://doi.org/10.1177/001316447003000105.

[165] Karen Kukich. "Techniques for automatically correcting words in text". In: *Acm Computing Surveys (CSUR)* 24.4 (1992), pp. 377–439.

[166] Matt J. Kusner et al. "From Word Embeddings to Document Distances". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015, pp. 957–966.

[167] William Labov. *Sociolinguistic patterns*. 4. University of Pennsylvania Press, 1972.

[168] L. Lamel et al. "The LIMSI Arise system". In: *Speech Communication* 31.4 (2000), pp. 339–353. ISSN: 0167-6393. DOI: https://doi.org/10.1016/S0167-6393(99)00067-9. URL: http://www.sciencedirect.com/science/article/pii/S0167639399000679.

[169] Guillaume Lample et al. "Multiple-Attribute Text Rewriting". In: *International Conference on Learning Representations*. 2019.

[170] Weiyu Lan, Xirong Li, and Jianfeng Dong. "Fluency-Guided Cross-Lingual Image Captioning". In: *Proceedings of the 2017 ACM on Multimedia Conference - MM '17* (2017). DOI: 10.1145/3123266.3123366. URL: http://dx.doi.org/10.1145/3123266.3123366.

[171] W. S. Lasecki, L. Rello, and J. P. Bigham. "Measuring text simplification with the crowd". In: *Proceedings of the 12th Web for All Conference (W4A '15)*. 2015.

[172] Alon Lavie and Abhaya Agarwal. "METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments". In: *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 228–231. URL: https://www.aclweb.org/anthology/W07-0734.

[173] Yann LeCun, Yoshua Bengio, et al. "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[174] Chris van der Lee et al. "Best practices for the human evaluation of automatically generated text". In: *Proceedings of the 12th International Conference on Natural Language Generation*. Tokyo, Japan: Association for Computational Linguistics, Oct. 2019, pp. 355–368. DOI: 10.18653/v1/W19-8643. URL: https://www.aclweb.org/anthology/W19-8643.

[175] Leo Leppänen et al. "Data-Driven News Generation for Automated Journalism". In: *Proceedings of the 10th International Conference on Natural Language Generation*. 2017, pp. 188–197.

[176] Mike Lewis et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension". In: *arXiv preprint arXiv:1910.13461* (2019).

[177] Jiwei Li et al. "A persona-based neural conversation model". In: *arXiv preprint arXiv:1603.06155* (2016).

[178] Juncen Li et al. "Delete, retrieve, generate: A simple approach to sentiment and style transfer". In: *arXiv preprint arXiv:1804.06437* (2018).

[179] Nannan Li and Zhenzhong Chen. *Learning Compact Reward for Image Captioning*. 2020. arXiv: 2003.10925 [cs.CV].

[180] Zhongyang Li, Xiao Ding, and Ting Liu. "Generating reasonable and diversified story ending using sequence to sequence model with adversarial training". In: *Proceedings of the 27th International Conference on Computational Linguistics*. 2018, pp. 1033–1043.

[181] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://www.aclweb.org/anthology/W04-1013.

[182] Chia-Wei Liu et al. "How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016, pp. 2122–2132.

[183] Ding Liu and Daniel Gildea. "Syntactic Features for Evaluation of Machine Translation". In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 25–32. URL: https://www.aclweb.org/anthology/W05-0904.

[184] Peter J Liu et al. "Generating wikipedia by summarizing long sequences". In: *arXiv preprint arXiv:1801.10198* (2018).

[185] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[186] Chi-kiu Lo. "MEANT 2.0: Accurate semantic MT evaluation for any output language". In: *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 589–597. DOI: 10.18653/v1/W17-4767. URL: https://www.aclweb.org/anthology/W17-4767.

[187] Chi-kiu Lo. "YiSi - a Unified Semantic MT Quality Evaluation and Estimation Metric for Languages with Different Levels of Available Resources". In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 507–513. DOI: 10.18653/v1/W19-5358. URL: https://www.aclweb.org/anthology/W19-5358.

[188] Chi-kiu Lo, Anand Karthik Tumuluru, and Dekai Wu. "Fully Automatic Semantic MT Evaluation". In: *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 243–252. URL: https://www.aclweb.org/anthology/W12-3129.

[189] Lajanugen Logeswaran and Honglak Lee. *An efficient framework for learning sentence representations*. 2018. arXiv: 1803.02893 [cs.CL].

[190] Jordan J. Louviere, Terry N. Flynn, and A. A. J. Marley. *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press, 2015. DOI: 10.1017/CBO9781107337855.

[191] Ryan Lowe et al. "Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2017). DOI: 10.18653/v1/p17-1103. URL: http://dx.doi.org/10.18653/v1/P17-1103.

[192] Hans Peter Luhn. "A statistical approach to mechanized encoding and searching of literary information". In: *IBM Journal of research and development* 1.4 (1957), pp. 309–317.

[193] Fuli Luo et al. "A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 5116–5122. URL: https://www.ijcai.org/Proceedings/2019/0711.pdf.

[194] Fuli Luo et al. "A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer". In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2019, pp. 5116–5122.

[195] Minh-Thang Luong et al. "Multi-task sequence to sequence learning". In: *arXiv preprint arXiv:1511.06114* (2015).

[196] V. Lyding et al. "The PAISÀ Corpus of Italian Web Texts". In: *Proceedings of the 9th Web as Corpus Workshop (WAC-9) EACL*. 2014, pp. 36–43.

[197] Iain Macdonald and Advaith Siddharthan. “Summarising news stories for children”. In: *Proceedings of the 9th International Natural Language Generation conference*. 2016, pp. 1–10.

[198] Isa Maks et al. “Generating Polarity Lexicons with WordNet propagation in five languages”. In: *Proceedings of LREC2014, Reykjavik* (2014).

[199] S. Malmasi et al. “A Report on the 2017 Native Language Identification Shared Task”. In: *Proceedings of the 12th Workshop on Building Educational Applications Using NLP*. 2017.

[200] Shervin Malmasi, Joel Tetreault, and Mark Dras. “Oracle and Human Baselines for Native Language Identification”. In: *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 172–178. DOI: 10.3115/v1/W15-0620. URL: https://www.aclweb.org/anthology/W15-0620.

[201] Gary Marcus. “Deep Learning: A Critical Appraisal”. In: *Computing Research Repository* abs/1801.00631 (2018). version 2. arXiv: 1801.00631. URL: http://arxiv.org/abs/1801.00631.

[202] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. “Generating Typed Dependency Parses from Phrase Structure Parses”. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-2006)*. 2006, pp. 449–454.

[203] Aleksandra Maslennikova et al. “Quanti anni hai? Age Identification for Italian”. In: *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019)*. Bari, Italy: CEUR Proceedings 2481, 2019.

[204] Philip Mccarthy and Scott Jarvis. “MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment”. In: *Behavior research methods* 42 (May 2010), pp. 381–92. DOI: 10.3758/BRM.42.2.381.

[205] R. McDonald et al. “Universal dependency annotation for multilingual parsing”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. 2013, pp. 92–97.

[206] Ryan McDonald, Koby Crammer, and Fernando Pereira. “Online large-margin training of dependency parsers”. In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*. 2005, pp. 91–98.

[207] M. Miestamo. “Grammatical complexity in a cross–linguistic perspective”. In: *M. Miestamo, K. Sinnemaki and F. Karlsson (eds.), Language Complexity: Typology, Contact, Change* Amsterdam: Benjamins (2008), pp. 23–41.

[208] Rada Mihalcea and Paul Tarau. “Textrank: Bringing order into text”. In: *Proceedings of the 2004 conference on empirical methods in natural language processing*. 2004.

[209] Tomáš Mikolov. “Statistical language models based on neural networks”. In: *Presentation at Google, Mountain View, 2nd April* 80 (2012).

[210] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[211] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[212] Jim Miller and Regina Weinert. "Spontaneous spoken language". In: *Syntax and discourse. Oxford, Clarendon Press* (1998).

[213] Scott Miller, Jethran Guinness, and Alex Zamanian. "Name tagging with word clusters and discriminative training". In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*. 2004, pp. 337–342.

[214] Remi Mir et al. "Evaluating Style Transfer for Text". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. June 2019, pp. 495–504.

[215] Tom M Mitchell et al. "Machine learning. 1997". In: *Burr Ridge, IL: McGraw Hill* 45.37 (1997), pp. 870–877.

[216] T. Mitra, C. Hutto, and E. Gilbert. "Comparing Person- and Process-centric Strategies for Obtaining Quality Data on Amazon Mechanical Turk". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015).

[217] Martin Molina, Amanda Stent, and Enrique Parodi. "Generating automated news to explain the meaning of sensor data". In: *International Symposium on Intelligent Data Analysis*. Springer. 2011, pp. 282–293.

[218] R. Munro et al. "Crowdsourcing and language studies: The new generation of linguistic data". In: *Proceedings of the Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. 2010, pp. 122–130.

[219] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *ICML*. 2010.

[220] Masami Nakamura and M Shikano. "A study of English word category prediction based on neutral networks". In: *International Conference on Acoustics, Speech, and Signal Processing,* IEEE. 1989, pp. 731–734.

[221] Ani Nenkova and Rebecca Passonneau. "Evaluating Content Selection in Summarization: The Pyramid Method". In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2004, pp. 145–152. URL: https://www.aclweb.org/anthology/N04-1019.

[222] Austin Lee Nichols and J. Maner. "The Good-Subject Effect: Investigating Participant Demand Characteristics". In: *The Journal of General Psychology* 135 (2008), pp. 151–166.

[223] J. Nivre et al. "The CoNLL 2007 Shared Task on Dependency Parsing". In: *Proceedings of the EMNLP-CoNLL*. 2007, pp. 915–932.

[224] Joakim Nivre, Johan Hall, and Jens Nilsson. "Maltparser: A data-driven parser-generator for dependency parsing." In: *LREC*. Vol. 6. 2006, pp. 2216–2219.

[225] Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. "RankME: Reliable Human Ratings for Natural Language Generation". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 72–78. DOI: 10.18653/v1/N18-2012. URL: https://www.aclweb.org/anthology/N18-2012.

[226] Jekaterina Novikova et al. "Why We Need New Evaluation Metrics for NLG". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*. Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 2241–2252. DOI: 10.18653/v1/D17-1237.

[227] Mick O'DONNELL et al. "ILEX: an architecture for a dynamic hypertext generation system". In: *Natural Language Engineering* 7.3 (2001), pp. 225–250.

[228] Franz Josef Och and Hermann Ney. "A systematic comparison of various statistical alignment models". In: *Computational linguistics* 29.1 (2003), pp. 19–51.

[229] M. Orne. "On the social psychology of the psychological experiment: With particular reference to demand characteristics and their implications." In: *American Psychologist* 17 (1962), pp. 776–783.

[230] Pedro Javier Ortiz Suárez, Laurent Romary, and Benoıt Sagot. "A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 1703–1714. URL: https://www.aclweb.org/anthology/2020.acl-main.156.

[231] Kishore Papineni et al. "BLEU: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[232] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the Difficulty of Training Recurrent Neural Networks". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. Atlanta, GA, USA: JMLR.org, 2013, III–1310–III–1318.

[233] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[234] Matthew E. Peters et al. "Deep contextualized word representations". In: *Proc. of NAACL*. 2018.

[235] Vassilis Plachouras et al. "Interacting with financial data using natural language". In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM. 2016, pp. 1121–1124.

[236] Kapila Ponnamperuma et al. "Tag2Blog: Narrative generation from satellite tag data". In: *Proceedings of the 51st annual meeting of the association for computational linguistics: System demonstrations*. 2013, pp. 169–174.

[237] Maja Popović et al. "Evaluation without references: IBM1 scores as evaluation metrics". In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, July 2011, pp. 99–103. URL: https://www.aclweb.org/anthology/W11-2109.

[238] François Portet et al. "Automatic generation of textual summaries from neonatal intensive care data". In: *Artificial Intelligence* 173.7-8 (2009), pp. 789–816.

[239] Peter Potash, Alexey Romanov, and Anna Rumshisky. "Ghostwriter: Using an lstm for automatic rap lyric generation". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 1919–1924.

[240] Martin Potthast et al. "A Stylometric Inquiry into Hyperpartisan and Fake News". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 231–240. DOI: 10.18653/v1/P18-1022. URL: https://www.aclweb.org/anthology/P18-1022.

[241] Shrimai Prabhumoye et al. "Style transfer through back-translation". In: *arXiv preprint arXiv:1804.09000* (2018).

[242] Alec Radford et al. *Improving language understanding by generative pre-training*. 2018.

[243] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI Blog* 1.8 (2019), p. 9.

[244] Pranav Rajpurkar et al. "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. DOI: 10.18653/v1/D16-1264. URL: https://www.aclweb.org/anthology/D16-1264.

[245] Alejandro Ramos-Soto et al. "Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data". In: *IEEE Transactions on Fuzzy Systems* 23.1 (2015), pp. 44–57.

[246] Sudha Rao and Joel Tetreault. "Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer". In: *arXiv preprint arXiv:1803.06535* (2018).

[247] Sravana Reddy and Kevin Knight. "Obfuscating gender in social media writing". In: *Proceedings of the First Workshop on NLP and Computational Social Science*. 2016, pp. 17–26.

[248] Radim Řehůřek and Petr Sojka. "Software Framework for Topic Modelling with Large Corpora". English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http://is.muni.cz/publication/884893/en. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[249] Katharina Reinecke and Krzysztof Z. Gajos. "LabintheWild: Conducting Large-Scale Online Experiments With Uncompensated Samples". In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work amp; Social Computing*. CSCW '15. Vancouver, BC, Canada: Association for Computing Machinery, 2015, pp. 1364–1378. ISBN: 9781450329224. DOI: 10.1145/2675133.2675246. URL: https://doi.org/10.1145/2675133.2675246.

[250] Ehud Reiter. "A Structured Review of the Validity of BLEU". In: *Computational Linguistics* 44.3 (2018), pp. 393–401. DOI: 10.1162/COLI.

[251] Ehud Reiter and Anja Belz. "An Investigation into the Validity of Some Metrics for Automatically Evaluating Natural Language Generation Systems". In: *Computational Linguistics* 35.4 (2009), pp. 529–558. DOI: 10.1162/coli.2009.35.4.35405. URL: https://www.aclweb.org/anthology/J09-4008.

[252] Ehud Reiter and Robert Dale. *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press, 2000. DOI: 10.1017/CBO9780511519857.

[253] Ehud Reiter, Roma Robertson, and Liesl M. Osman. "Lessons from a failure: Generating tailored smoking cessation letters". In: *Artificial Intelligence* 144.1 (2003), pp. 41–58. ISSN: 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(02)00370-3. URL: http://www.sciencedirect.com/science/article/pii/S0004370202003703.

[254] Ehud Reiter and Somayajulu Sripada. "Should Corpora Texts Be Gold Standards for NLG?" In: *Proceedings of the International Natural Language Generation Conference*. Harriman, New York, USA: Association for Computational Linguistics, July 2002, pp. 97–104. URL: https://www.aclweb.org/anthology/W02-2113.

[255] Ehud Reiter et al. "Choosing words in computer-generated weather forecasts". In: *Artificial Intelligence* 167.1-2 (2005), pp. 137–169.

[256] Jason D. M. Rennie and Ryan Rifkin. "Improving Multiclass Text Classification with the Support Vector Machine". In: *MIT AI Memos (1959 - 2004)* (2001).

[257] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1278–1286. URL: http://proceedings.mlr.press/v32/rezende14.html.

[258] Brian Richards. "Type/Token Ratios: what do they really tell us?" In: *Journal of child language* 14 (July 1987), pp. 201–9. DOI: 10.1017/S0305000900012885.

[259] Brian Roark et al. "Deriving lexical and syntactic expectation-based measures for psycholinguistic modeling via incremental top–down parsing". In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2009, pp. 324–333.

[260] Stephen E Robertson and K Sparck Jones. "Relevance weighting of search terms". In: *Journal of the American Society for Information science* 27.3 (1976), pp. 129–146.

[261] Martina Astrid Rodda, Marco SG Senaldi, and Alessandro Lenci. "Panta Rei: Tracking Semantic Change with Distributional Semantics in Ancient Greek." In: *CLiC-it/EVALITA*. 2016.

[262] Sebastian Ruder et al. "Sluice networks: Learning what to share between loosely related tasks". In: *arXiv preprint arXiv:1705.08142* (2017).

[263] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[264] Vasile Rus et al. "The first question generation shared task evaluation challenge". In: *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics. 2010, pp. 251–257.

[265] Alexander M Rush, Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization". In: *arXiv preprint arXiv:1509.00685* (2015).

[266] K. Sagae, A. Lavie, and B. MacWhinne. "Automatic measurement of syntactic development in child language". In: *Proceedings of the 43rd Annual Meeting of the ACL*. 2005.

[267] Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. *A Survey of Evaluation Metrics Used for NLG Systems*. 2020. arXiv: 2008.12009 [cs.CL].

[268] Abhilasha Sancheti et al. "Reinforced Rewards Framework for Text Style Transfer". In: *Advances in Information Retrieval*. 2020, pp. 545–560.

[269] Tobias Schnabel et al. "Evaluation methods for unsupervised word embeddings". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 298–307.

[270] Mike Schuster and Kuldip K Paliwal. "Bidirectional recurrent neural networks". In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[271] Holger Schwenk and Jean-Luc Gauvain. "Training neural network language models on very large corpora". In: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2005, pp. 201–208.

[272] Abigail See, Peter J. Liu, and Christopher D. Manning. "Get To The Point: Summarization with Pointer-Generator Networks". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1073–1083. DOI: 10.18653/v1/P17-1099. URL: https://www.aclweb.org/anthology/P17-1099.

[273] Thibault Sellam, Dipanjan Das, and Ankur Parikh. "BLEURT: Learning Robust Metrics for Text Generation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 7881–7892. DOI: 10.18653/v1/2020.acl-main.704. URL: https://www.aclweb.org/anthology/2020.acl-main.704.

[274] Rico Sennrich, Barry Haddow, and Alexandra Birch. "Neural machine translation of rare words with subword units". In: *arXiv preprint arXiv:1508.07909* (2015).

[275] Aliaksei Severyn and Alessandro Moschitti. "UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 464–469. DOI: 10.18653/v1/S15-2079. URL: https://www.aclweb.org/anthology/S15-2079.

[276] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[277] Naeha Sharif et al. "Learning-based Composite Metrics for Improved Caption Evaluation". In: *Proceedings of ACL 2018, Student Research Workshop*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 14–20. DOI: 10.18653/v1/P18-3003. URL: https://www.aclweb.org/anthology/P18-3003.

[278] Advaith Siddharthan. "A survey of research on text simplification". In: *ITL-International Journal of Applied Linguistics* 165.2 (2014), pp. 259–298.

[279] Maria Simi, Cristina Bosco, and Simonetta Montemagni. "Less is More? Towards a Reduced Inventory of Categories for Training a Parser for the Italian Stanford Dependencies". In: *Proceedings of the 9th International Conference on Language Resources and Evaluation, (LREC'14)*. 2014, pp. 83–90.

[280] Matthew Snover et al. "A Study of Translation Edit Rate with Targeted Human Annotation". In: *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*. Cambridge, Massachusetts, USA: Association for Machine Translation in the Americas, Aug. 2006, pp. 223–231. URL: https://www.aclweb.org/anthology/2006.amta-papers.25.

[281] Anders Søgaard and Yoav Goldberg. "Deep multi-task learning with low level tasks supervised at lower layers". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2. 2016, pp. 231–235.

[282] Alessandro Sordoni et al. "A neural network approach to context-sensitive generation of conversational responses". In: *arXiv preprint arXiv:1506.06714* (2015).

[283] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[284] A. Staub et al. "Distributional effects of word frequency on eye fixation durations". In: *Journal of Experimental Psychology: Human Perception and Performance* 36 (2010), pp. 1280–1293.

[285] Oliviero Stock et al. "Adaptive, intelligent presentation of information for the museum visitor in PEACH". In: *User Modeling and User-Adapted Interaction* 17.3 (2007), pp. 257–304.

[286] C. Sun et al. "VideoBERT: A Joint Model for Video and Language Representation Learning". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 7463–7472. DOI: 10.1109/ICCV.2019.00756.

[287] Fan-Keng Sun and Cheng-I Lai. "Conditioned Natural Language Generation using only Unconditioned Language Model: An Exploration". In: *ArXiv* abs/2011.07347 (2020).

[288] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*. 2014, pp. 3104–3112.

[289] Rachael Tatman. "Evaluating Text Output in NLP: BLEU at your own risk". In: *Web: https://towardsdatascience. com/evaluating-text-outputin-nlp-bleu-at-your-own-risk-e8609665a213* (2019).

[290] Yi Tay et al. "Efficient transformers: A survey". In: *arXiv preprint arXiv:2009.06732* (2020).

[291] Mariët Theune et al. "From data to speech: a general approach". In: *Natural Language Engineering* 7.1 (2001), pp. 47–86.

[292] Youzhi Tian, Zhiting Hu, and Zhou Yu. "Structured Content Preservation for Unsupervised Text Style Transfer". In: *arXiv preprint arXiv:1810.06526* (2018).

[293] Alexey Tikhonov and Ivan P Yamshchikov. "Guess who? Multilingual approach for the automated generation of author-stylized poetry". In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2018, pp. 787–794.

[294] Ross Turner et al. "Using spatial reference frames to generate grounded textual summaries of georeferenced data". In: *Proceedings of the fifth international natural language generation conference*. Association for Computational Linguistics. 2008, pp. 16–24.

[295] Peter D Turney and Patrick Pantel. "From frequency to meaning: Vector space models of semantics". In: *Journal of artificial intelligence research* 37 (2010), pp. 141–188.

[296] Arjen Van Dalen. "The algorithms behind the headlines: How machine-written news redefines the core skills of human journalists". In: *Journalism Practice* 6.5-6 (2012), pp. 648–658.

[297] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[298] James Vincent. "OpenAI has published the text-generating AI it said was too dangerous to share". In: *The Verge* (Nov. 7, 2019). URL: https://www.theverge.com/2019/11/7/20953040/openai-text-generation-ai-gpt-2-full-model-release-1-5b-parameters (visited on 02/17/2021).

[299] René van der Wal et al. "The role of automated feedback in training and retaining biological recorders for citizen science". In: *Conservation Biology* 30.3 (2016), pp. 550–561.

[300] Sinong Wang et al. "Linformer: Self-Attention with Linear Complexity". In: *arXiv preprint arXiv:2006.04768* (2020).

[301] Leo Wanner et al. "Getting the environmental information across: from the web to the user". In: *Expert Systems* 32.3 (2015), pp. 405–432.

[302] Ronald Wardhaugh. *An introduction to sociolinguistics*. Vol. 28. John Wiley & Sons, 2011.

[303] Sean Welleck et al. *Neural Text Generation with Unlikelihood Training*. 2019. arXiv: 1908.04319 [cs.LG].

[304] Tsung-Hsien Wen et al. "Semantically conditioned lstm-based natural language generation for spoken dialogue systems". In: *arXiv preprint arXiv:1508.01745* (2015).

[305] Paul J Werbos. "Backpropagation through time: what it does and how to do it". In: *Proceedings of the IEEE* 78.10 (1990), pp. 1550–1560.

[306] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis". In: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. HLT '05. Vancouver, British Columbia, Canada: Association for Computational Linguistics, 2005, pp. 347–354. DOI: 10.3115/1220575.1220619. URL: https://doi.org/10.3115/1220575.1220619.

[307] Sam Wiseman, Stuart Shieber, and Alexander Rush. "Challenges in Data-to-Document Generation". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2253–2263. DOI: 10.18653/v1/D17-1239. URL: https://www.aclweb.org/anthology/D17-1239.

[308] Hiroyasu Yamada and Yuji Matsumoto. "Statistical dependency analysis with support vector machines". In: *Proceedings of the Eighth International Conference on Parsing Technologies*. 2003, pp. 195–206.

[309] Qian Yang, Rebecca J. Passonneau, and Gerard de Melo. "PEAK: Pyramid Evaluation via Automated Knowledge Extraction". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16. Phoenix, Arizona: AAAI Press, 2016, pp. 2673–2679.

[310] Mary Lynn Young and Alfred Hermida. "From Mr. and Mrs. outlier to central tendencies: Computational journalism and crime reporting at the Los Angeles Times". In: *Digital Journalism* 3.3 (2015), pp. 381–397.

[311] R. Michael Young. "Using Grice's Maxim of Quantity to Select the Content of Plan Descriptions". In: *Artif. Intell.* 115.2 (Dec. 1999), pp. 215–256. ISSN: 0004-3702. DOI: 10.1016/S0004-3702(99)00082-X. URL: https://doi.org/10.1016/S0004-3702(99)00082-X.

[312] Hui Yu et al. "RED: A Reference Dependency Based MT Evaluation Metric". In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 2042–2051. URL: https://www.aclweb.org/anthology/C14-1193.

[313] Rowan Zellers et al. "Defending against neural fake news". In: *Advances in Neural Information Processing Systems*. 2019, pp. 9051–9062.

[314] Shiwei Zhang et al. "Irony detection via sentiment-based transfer learning". In: *Information Processing Management* 56.5 (2019), pp. 1633–1644. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2019.04.006. URL: https://www.sciencedirect.com/science/article/pii/S0306457318307428.

[315] Yuhao Zhang et al. "Optimizing the Factual Correctness of a Summary: A Study of Summarizing Radiology Reports". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020). DOI: 10.18653/v1/2020.acl-main.458. URL: http://dx.doi.org/10.18653/v1/2020.acl-main.458.

[316] Tianyi Zhang* et al. "BERTScore: Evaluating Text Generation with BERT". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=SkeHuCVFDr.

[317] Li Zhou et al. "The Design and Implementation of XiaoIce, an Empathetic Social Chatbot". In: *Computational Linguistics* 46.1 (Mar. 2020), pp. 53–93. ISSN: 1530-9312. DOI: 10.1162/coli_a_00368. URL: http://dx.doi.org/10.1162/coli_a_00368.

[318] Yaoming Zhu et al. "Texygen: A Benchmarking Platform for Text Generation Models". In: *The 41st International ACM SIGIR Conference on Research amp; Development in Information Retrieval*. SIGIR '18. Ann Arbor, MI, USA: Association for Computing Machinery, 2018, pp. 1097–1100. ISBN: 9781450356572. DOI: 10.1145/3209978.3210080. URL: https://doi.org/10.1145/3209978.3210080.

[319] Daniel M. Ziegler et al. “Fine-Tuning Language Models from Human Preferences”. In: *CoRR* abs/1909.08593 (2019). arXiv: 1909.08593. URL: http://arxiv.org/abs/1909.08593.