UNIVERSITÀ DI PISA

Dipartimento di Filologia, Letteratura e Linguistica

Corso di Laurea Magistrale in Informatica Umanistica

TESI DI LAUREA

2022

Probing of Pre-Trained Language Models for Metonymy Classification: a new Dataset and Experiments

Relatore:

Prof. Alessandro Lenci

Candidata:

Chiara Fazzone

A mio fratello

Abstract

Pre-trained language models such as BERT (Bidirectional Encoder Representations from Transformers) have achieved state of the art performances in NLP. The success of such models is due to the contextualized vector representations of language, also known as embeddings, they are able to generate and that have attracted much attention from researchers. In fact, pre-trained language models suffer from low interpretability, meaning that is it difficult to identify the pieces of information encoded in their embeddings, and where and how they are encoded. The aim of this work is to probe the vector representations extracted from BERT to understand whether it captures some information related to one linguistic phenomenon in particular, metonymy.

Table of Contents

«The complete meaning of a word is always contextual,

and no study of meaning apart from context can be taken seriously»

(John Rupert Firth)

1. Introduction

Natural Language Processing (NLP) applications have achieved high levels of efficiency and accuracy on a number of natural-language-related tasks, mainly due to the exceptional quality of the language representations some models are able to create. With the growing success and interest in the deep neural networks responsible for such achievements, comes an equally growing curiosity towards what makes these performances possible. In fact, these deep language models, often belonging to the family of the Transformers (Vaswani et al., 2017), create vector representations of language that are not interpretable by humans. The need to understand how they carry out predictions and why they make some choices instead of others has started a series of research explorations, known as *probing tasks*, of their representations aimed at identifying what linguistic information they capture, how they encode it and where it is encoded in their embeddings.

The probing methodology consists in taking vector representations and performing a classification task with respect to some information. The idea is that the task can only be successful if the information at issue is present in the representation and is available for the classifier to use to distinguish different classes. This has been done for many aspects of natural language, from text

structure to syntactic and semantic information (see Section 2.2. for *probing tasks*). It is particularly interesting to explore to what extent and how language models capture information related to sematic composition, that is to say how the meaning of words changes when these are combined with other context words.

The aim of this work is to find whether pre-trained language models are able to capture and encode the meaning shift brought by metonymy (Section 3.), a linguistic phenomenon for which meaning changes in context and that occurs when the name of an entity is used to refer to another entity to which it is in some way closely related. To do so, a new dataset was created that includes examples for a variety of occurrences of metonymy. The vector representations of the sentences in the dataset are extracted from a pre-trained language model and given as input to the probing classifier.

Each different phase of the work, from the study of literature on metonymy to the creation of the dataset and the results of the probing task, finds a corresponding section in this thesis: Section 2. provides an introduction to the state of the art in NLP, in particular to word and sentence embeddings (Section 2.1.) and probing tasks (Section 2.2.); Section 3. describes metonymy (Section 3.1.) and the existing datasets for the linguistic phenomenon (Sections 3.2. and 3.3.); Section 4. retraces the creation of a new dataset for metonymy, specifically

the types of metonymy included (Section 4.1.), the collection and annotation (Section 4.2.) and the validation of data (Section 4.3.), the collection and annotation of negative examples (Section 4.4.) and the complete dataset statistics (Section 4.5.); Section 5 is dedicated to the experiments, with a brief introduction to the pre-trained language model and the probing classifier (Section 5.1.), and a description of the different test sets (Section 5.2.), the probing classification task (Section 5.3.), the performances of the classifier (Section 5.4.) and the analysis of results (Section 5.5.).

2. State of the Art

In the last years, the field of Natural Language Processing (NLP) has seen many exciting progresses that have captured the attention of the whole Artificial Intelligence (AI) community. The availability of big amounts of textual data, as well as an approach that exploits probabilistic and statical methods, has played a fundamental role in supporting the growth of these disciplines (Lenci et al., 2016). Johnson (2009) observes how the statistical revolution has influenced change in Computational Linguistics in the last decades of the twentieth century, leading the process that substituted a grammar-based approach, which used a set of rules or grammars manually defined, with a model-based one, which relies on probabilistic and statistical methods.

The aim of this section is to provide an introduction to the state-of-the-art (SoTA) panorama for NLP, presenting the models that make full use of both the large availability of data and statistical methods, and that have reached impressive performances in many NLP tasks. Section 2.1. focuses in particular on word and sentence embeddings, the fundamental information-bearing components for any NLP task; Section 2.2. describes their interpretability issues and a method to explore them.

2.1. Word and sentence embeddings

Distributional semantics (DS) is a usage-based model of meaning that provides multi-dimensional and graded word representations that capture many aspects of meaning in natural language (Lenci, 2018; Boleda, 2020). The idea is that the semantic behaviour of linguistic items is highly dependent on their statistical distribution in context. In fact, distributional models create semantic representations for words using their co-occurrences extracted from corpora. The model is based on the Distributional Hypothesis, which lies on the observation that similarity in meaning corresponds to similarity in linguistic distribution (Harris, 1954).

A distributional representation is a vector representing the co-occurrence of a word with linguistic contexts. Words that occur in similar contexts are expected to have similar meanings and are therefore close in the vector space. For example, the representations for *cat* and *dog* are expected to be closer in the distributional space than the vectors for *cat* and *truck*.

Distributional semantics builds upon the vector space model in information retrieval (Salton et al., 1975), in which a collection of documents is represented as a matrix whose columns are vectors corresponding to documents and rows are vectors corresponding to lexical items, and in which each matrix entry

registers the occurrences of a word in a document. Distributional Semantic Models, instead, project text in a vector space where each word is represented as a *word embedding*, a vector whose components encode the distribution of the word in the different contexts (Lenci, 2018).

Word embeddings are a key ingredient to carry out any NLP task, the performance of which depends on the quality of such representations. These can be built using co-occurrence matrices, as seen above, or neural networks able to extract features from the text they are encoding in un unsupervised way. Deep neural networks, so called because of their architecture composed of multiple hidden layers, can obtain pre-trained word embeddings that can be exploited in other NLP tasks by autonomously extracting features from the corpora used in the training phase.

The efficiency of static word embeddings extracted by distributional models based on unsupervised algorithms such as Word2Vec (Mikolov et al; 2013), GloVe (Pennington et al.; 2014) or FastText (Bojanowski et al.; 2017) was questioned due to their inadequacy to capture semantic polysemy or meaning shifts, which occurs when a word takes different meanings in different contexts. Distributional semantic models, indeed, represent the meaning of a lexical item through a single vector representation that in some sense compresses its whole

distributional history. In other words, all senses of a polysemous word must share a single vector.

The recent introduction of deep neural architectures for language modelling has captured great interest for the SoTA results achieved in many NLP task thanks to the word representations they are able to create. These are contextualized word embeddings, as the ones created by ELMo (Embedding from Language Models; Peters et al., 2018), GPT (Generative Pre-Training, Radford et al., 2018) and its later versions GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020), and BERT (Bidirectional Encoder Representations from Transformers, Devlin et al., 2019), deep neural language models that are fine-tuned to create models for a number of NLP tasks. These models compute dynamic word embeddings for words given their context sentence, and therefore largely address polysemy. Contextualized word representations are context-sensitive and generally perform better than static ones on tasks related to lexical composition and meaning shift (Shwartz and Dagan, 2019).

ELMo creates deep context-dependent representations of each token by concatenating the internal states of a 2-layer biLM (bidirectional language model) pre-trained on a large text corpus (Peters et al., 2018). Unlike traditional word embeddings as the ones presented above, the ELMo vector for a word is a function of the entire sentence it is contained in, which means that in different

contexts the same word can have different representations. BERT and GPT-2 are bi-directional and uni-directional transformer-based language models respectively. Much of the recent progress in NLP can be attributed to a neural network architecture called *Transformer*, a model architecture that relies on the attention mechanism introduced in the paper 'Attention is all you need' (Vaswani et al., 2017). A Transformer has an encoder-decoder structure in which the attention mechanism is used to pass a more complete picture of the whole sequence rather than one element at the time to the decoder. Contextualized word representations of each token are created by each transformer layer by attending to different parts of the input sentence (Devlin et al., 2019; Radford at al., 2019).

However, language representations also have to account for issues related to semantic compositionality, which is the human ability to compose lexical meanings to create a potentially unlimited number of complex linguistic expressions (Lenci, 2018). The most common compositional approach in DS is to use linear-algebraic operations to project lexical vectors to phrase vectors. Vector addition (Landauer and Dumais, 1997) is the simplest form of vector composition. Simple additive and multiplicative methods are better than more complex distributional methods for semantic compositionality (Blacoe and Lapata 2012), even though they are still not fully adequate as vector addition

models are unable to account for syntax and multiplicative models make no distinction between the constituents they combine (Mitchell and Lapata, 2010).

Other approaches use unsupervised neural models such as Skip Thought (Kiros et al., 2015), an encoder-decoder model. It abstracts the Skip-Gram model, an algorithm of Word2Vec (Mikolov et al; 2013) which uses a word to predict its surrounding context, to the sentence level and encodes a sentence to predict the sentences around it. The sentence embeddings generated by supervised algorithms were only recognized as valuable since the publication of a work by Conneau et al. (2017), when the InferSent method was presented. The latter was trained on the Stanford Natural Language Inference corpus (SNLI) (Bowman et al., 2015) and outperformed the Skip-Thought vectors, showing that models learned on natural language inference (NLI) can perform better than models trained in unsupervised conditions.

The Universal Sentence Encoder proposed by Google (Cer et al., 2018) combines the strategies of both Skip-Thought (Kiros et al., 2015) and InferSent (Conneau et al. (2017): the sentence-encoding model is based on a Transformer architecture (Vaswani et al., 2017) that uses attention to compute context-aware representations of words in a sentence. The multi-task learning includes a SkipThought like task for the unsupervised learning from arbitrary running text, and classification tasks for training on supervised data.

Recently, large pre-trained language models such as BERT (Devlin et al., 2018) have been used to extract representations of natural language, achieving a new state-of-the-art performance level on sentence-pair regression tasks like semantic textual similarity (STS) (Reimers and Gurevych, 2019). One of the methods implemented to extract sentence-level information is to use mean or maximum pooling, which is to consider the average or maximum value across each of the 512 dimensions in the hidden state embeddings. Another one is to select the first vector of the hidden state as the class token which represents the sequential information of the embedding sequence. This method will be used to extract the sentence embeddings for the work described in Section 5. The same can be done with GPT-2 (Radford et al., 2019), with the only difference that the sequential information is encoded in the last token of the embedding sequence because of the uni-directional self-attention mechanism.

2.2. Probing task

An issue with pretrained language models is their low interpretability, defined by Miller (2019) as the degree to which a human can understand the cause of a decision, and by Been et al. (2016) as the degree to which a human can consistently predict the model's result. Deep neural networks, which have

achieved near-human levels of accuracy in different type of prediction and classification tasks, are treated mostly as black-box function approximators that map a given input to a classification output (Chakraborty et al., 2017; Ribeiro et al., 2016). It is important, however, to provide human-understandable justifications for the neural networks' outputs that lead to insights about their inner workings. In fact, the dense representations of these models are still poorly understood. We have a limited understanding of the information they capture, of where it is encoded and of why they perform so well on many NLP tasks.

Contextual representations and attention weights have been interpreted mostly through *probes*, classifiers trained on said representations to predict a certain property. This framework is known as probing task and is a common methodology used to associate internal representations with external properties to answer questions about the structure of models. The idea is the following: a model is trained on some task, such as language modelling; the pre-trained representations generated using the model are given as input to another classifier that is trained to take the representations and predict some property. If the classifier performs well, it means that the model has learnt information that is relevant to the property (Belinkov, 2022). The probe should be a simple classifier so that the classification performance can be attributed to the quality

of the information encoded in the embeddings rather than to the ability of the classifier to find alternative patterns.

The difficulty to assess what information these representations are capturing is particularly acute for compositional meaning information. Several studies have tried to explain how language models encode the semantic meaning of sentences. Some examples, just to mention a few, are the works from Ettinger et al. (2016), Adi et al. (2016), Conneau et al. (2018), Ettinger et al. (2018), Shwartz and Dagan (2019).

Ettinger et al. (2016) and Ettinger et al. (2018) probe semantic evidence of compositionality through a classification task. Sentence-level meaning representations can be formed from word-level representations. In this case, to evaluate the sentence embedding it is central to evaluate how effectively the model has performed the composition process, defined as a generation of meaning that makes available all the information that we would expect to be able to extract from the input sentence. They propose methods to assess specific semantic information captured in sentence representations that involve the construction of a sentence dataset annotated for some linguistic characteristics and a test for extractability of sematic information by means of a simple classification task performed on the vector representations for the sentences on tasks defined by those target linguistic characteristics.

Adi et al. (2016) define prediction tasks for probing the encoding of sentence structure information in sentence representations, namely sentence length, word content and word order. They score the quality of the representations according to the ability of a classifier to solve such prediction tasks when the representations are given as input.

Conneau et al. (2018) use encoders pre-trained on a certain task to produce sentence embeddings. A classifier is then trained on these representations. They define a set of probing tasks organized by the type of linguistic properties being probed: surface, syntactic and sematic information. The first requires no linguistic knowledge and involves the prediction of the length of sentences in terms of number of words and the word content. The second set of tasks tests whether sentence embeddings are sensitive to syntactic properties of the sentences they encode and the third requires some understanding of what a sentence denotes.

Shwartz and Dagan (2019) deal with the meaning shift of constituent words in a phrase and the introduction of implicit information caused by lexical composition. They compare contextualized word embeddings with static word embeddings and address lexical composition through classification tasks.

In this thesis a new dataset (Section 4.) and experiments (Section 5.) will be described for the probing of pre-trained language models on metonymy (Section

3.) classification. The aim is to detect whether the information for the meaning shift operated by metonymy is easily traceable in sentence embeddings created from SoTA pre-trained language models. Following the probing strategy adopted in the studies mentioned above, the experiments will be run on sentence embeddings which are given as input to a simple classifier. We expect the performance of the classifier to reveal if the information relating to the meaning shift is directly encoded in the representations.

3. Metonymy

In the following sections, metonymy is introduced (Section 3.1.) as well as a short survey on the existing datasets on the phenomenon (Sections 3.2. and 3.3.) In the first part, different types of metonymies are described and illustrated with examples in order to have a clear understanding of how metonymy operates in natural language and how diverse its occurrences can be. In the second and the third, the different datasets for metonymy will be introduced and analysed to give some contextual background to the work discussed in the rest of the thesis with special regards to the creation of a new dataset (Section 4.).

3.1. What is metonymy?

The cognitive and linguistic process considered in this study is *metonymy*. It is a figure of speech in which meaning changes in context in specific ways: it occurs when the speaker uses the name of an entity to refer to another entity to which it is in some way closely related. It sounds perfectly natural to speakers to use, for example, the expression '*to drink a bottle*', where '*bottle*' refers to the liquid it contains and not the plastic or glass object containing the liquid. The entities involved can be related physically, casually, spatially, or according to other relations. In most cases the use of metonymy generates a semantic type

shift. Metonymy is sometimes seen as a kind of metaphor. Indeed, similarly to metaphor, which still allows to refer to one thing through another entity, but with a less obvious and more abstract relationship, metonymy is a productive mechanism that operates semantic and lexical change (Pustejovski and Batjukova, 2019).

Not all relationships between lexical items can be exploited to produce metonymies. While an arm and a leg are both parts of a body, and are therefore closely related in space, one cannot metonymically stand for the other, i.e. '*I broke my leg*' cannot be interpreted as '*I broke my arm*'. A relationship between entities is suited to form a metonymy when there is a conceptual distinctness between the two. In '*She was treated by an ambulance*', the conceptual contrast between the vehicle and the paramedics is evident (Radden and Kövecses, 1999).

One common use of metonymy is where the part stands for the whole, as in '*Most of those have to go upstairs and I'll need to hire* some muscle (= strong people) *to do that*'. Among the many parts a whole has, we pick the one that better describes the aspect we are focusing on. The sentence above could be modified as '*These projects have to be finished and I'll need to hire some* good heads (= intelligent people) *to do that*'. In both cases a part (muscle, head) stands

for a whole (person), but one part or the other is chosen according to the specific characteristic it is associated with.

The part-whole relationship between entities is not the only one that allows the production of metonymic expressions. Metonymic concepts are in some way systematic and their examples can be grouped according to the kind of relationship involved. According to Lakoff and Johnson (1980), metonymic relations can be described by the following taxonomy:

- *Part for whole*: 'I'll need to hire some muscle'.
- *Producer for product*: 'Smith was driving a Ford'.
- *Object used for user*: 'The piano will be late today'.
- *Controller for controlled*: 'Nixon bombed them anyway'.
- *Institution for people responsible*: 'The university will not agree'.
- *The place for the institution*: 'The Kremlin treats its own citizens with contempt'.
- *The place for the event*: 'We can ask about the health effects following Hiroshima'.

Other taxonomies exist, like the more complex one defined by Radden and Kövecses (1999) (Figure 1), where at least 16 key types of metonymies are identified.

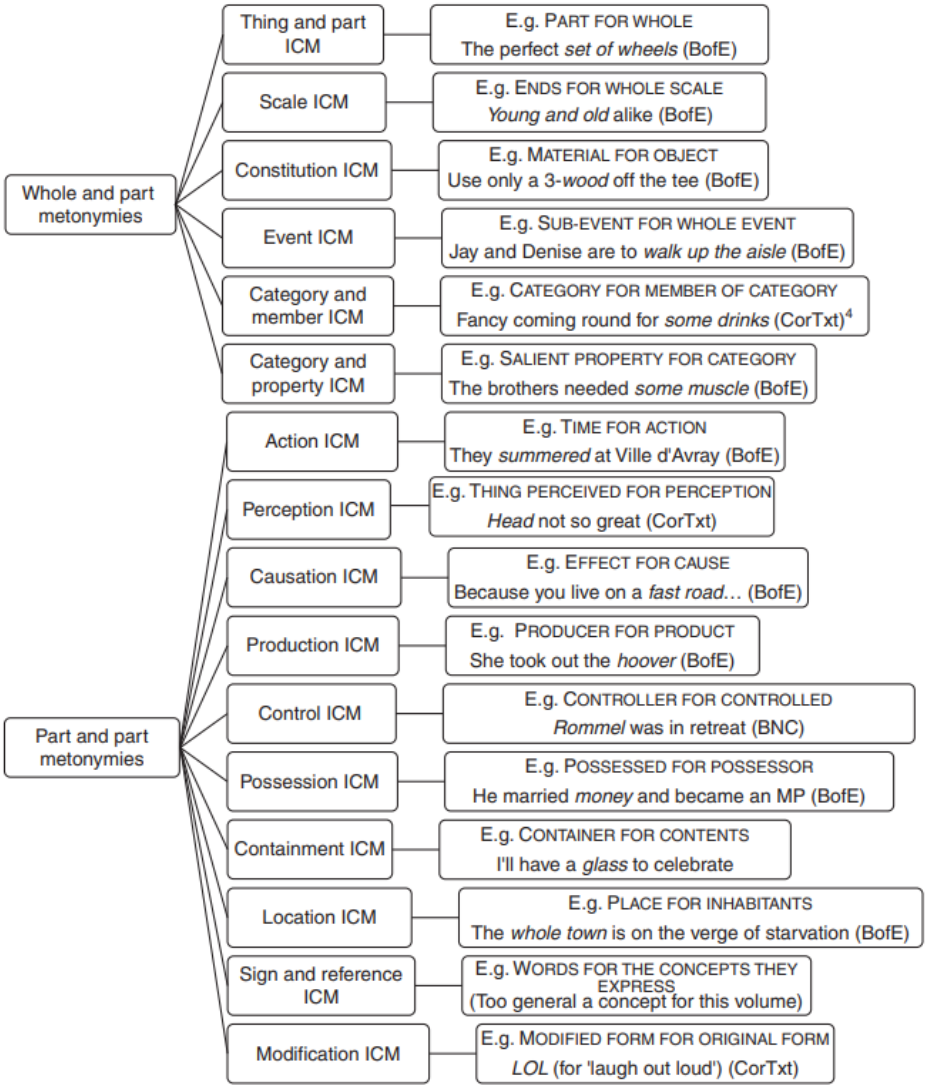

Figure 1: Key metonymy types in Radden and Kövecses' (1999) taxonomy. Image taken from Littlemore (2015).

Metonymic instances like the ones above are also examples of how we organize our thoughts. In fact, they allow us to conceptualize one thing according to its relation to something else. For example, in '*The piano will be late today*' the

speaker is not interested in the person as a person but only in the fact that he or she is the musician that specifically plays the piano. Moreover, the use and understanding of the metonymic process goes beyond language and intersects both cognition and culture. The identification of a person based on its face rather than on the rest of the body is a tradition that is based on metonymic conceptualization and is evident in art and photography as well. We usually recognize people by their face, which is a part of the whole body and what we expect to see if we ask for a picture of someone. We can also use simple concepts to refer to something more complex or abstract, so that '*Hiroshima*' can be used to refer to some events that took place in that site. In other words, speakers sometimes conceptualize and perceive things in metonymic terms (Lakoff and Johnson, 2008).

Furthermore, in order to understand metonymic expressions, we use our own knowledge of the world. We know that '*Hiroshima*' refers to the atomic bombing of Hiroshima of 1945 at the hands of the United States and that '*a Picasso*' is certainly a painting because Pablo Picasso was a painter, so the producer of the piece of art. At the same way we know that an ambulance is a vehicle and cannot treat people, but there are paramedics on board who can, and that we cannot drink a bottle but the liquid inside it. In some ways, metonymy is a communicative shorthand (Littlemore, 2015) that allows people to

communicate using their shared knowledge of the world and less words than they would otherwise need.

3.2. Existing datasets for metonymy resolution

The existing datasets on metonymy focus mainly on location names, so geographical territories or political entities. One very common instance of metonymy, indeed, is the use of the location name to refer to people, events or products. In the sentence '*Italy won the World Cup*', '*Italy*' refers to the football team; at the same way, in '*Nobody wants another Vietnam*', '*Vietnam*' stands for the Vietnam war, and in '*We would love some Bordeaux*' the region name stands for the wine produced there.

3.2.1. Metonymy resolution

According to Gritta et al. (2017), about 20% of location names in data sampled from Wikipedia are used metonymically. Resolving metonymy can improve many natural language processing tasks such as machine translation, question answering, named entity recognition, word sense disambiguation and coreference resolution. The task of metonymy resolution (MR) aims to identify

and appropriately interpret words that are used metonymically, that is to say to determine whether a potentially metonymic word (PMW) in a given context is used metonymically or not (Li et al. 2020). A number of different datasets were released by researchers for metonymy resolution, including SemEval (Nissim and Markert, 2007), ReLocaR (Gritta et al., 2017) and WiMCor (Kevin and Michael, 2020), which were mainly used for conventional approaches making extensive use of taggers, parsers, lexicons and hand-crafted or corpus-derived features (Nissim and Markert, 2003; Farkas et al., 2007; Nastase et al., 2012) and which will be described in the following sections.

3.2.2. An annotated dataset for locations

Nissim and Markert (2003) introduced an annotated corpus of occurrences of country names. The annotation scheme identifies literal, metonymic and mixed readings. *Literal* comprises a locative and a political entity interpretation ('*coral coast of Papua New Guinea*')[1]. *Metonymic* further specifies some patterns: *place-for-people*: a place stands for any people or organizations associated with it ('*England lost in the semi-final.*') ; *place-for-event*: a place stands for an event that occurred there ('*Sex, drugs, and Vietnam have haunted Bill Clinton's*

[1] Examples were taken from Nissim and Markert (2003).

campaign.'); *place-for-product*: a place stands for the product manufactured there ('*a smooth Bordeaux that was gutsy enough to cope with our food*'); *other met*: unconventional metonymies that do not fit in any of the other patterns ('*The thing about the record is the influences of the music. The bottom end is very New York/ New Jersey and the top is very melodic.*'). Finally, *mixed* is used when both interpretations are triggered by two different predicates and coexist in the same sentence ('*they arrived in Nigeria, hitherto a leading critic of [...]*').

1000 occurrences of country names, each surrounded by three sentences of context, were extracted from the British National Corpus (BNC) (BNC Consortium, 2007) and were annotated independently by the two authors. Only the examples both annotators would agree on were included in the corpus, and noisy elements were removed. The result is a corpus of 925 examples, of which 737 are literal and 188 are non-literal. In this study metonymy resolution is treated as a word sense disambiguation (WSD) task, as it can be reformulated as a classification task between the literal interpretation of the word and the set of possible metonymic patterns and is therefore concerned with distinguishing between possible word senses or readings.

3.2.3. SemEval 2007

The SemEval 2007 Shared Task 8 (Markert and Nissim, 2007) dataset still keeps the subtask concentrating on the semantic class *location* and adds another one concentrating on *organization*, exemplified by company names. The possible annotation categories defined by Nassim and Markert (2003) were maintained as well as the four metonymic patterns for *location*, while six metonymic patterns were included for the class *organization*: *org-for-members*, where the organization stands for its members, such as a spokesperson or its employees ('*Last February IBM announced [. . .]*'[2]); *org-for-event*, where it is used to refer to an event associated with the organization ('*The resignation of Leon Brittan from Trade and Industry in the aftermath of Westland.*'); *org-for-product*, where the name of a company refers to its products ('*His BMW went on to race at Le Mans*'); *org-for-facility*, where it stands for the facility housing the organization or one of its branches ('*The opening of a McDonald's is a major event*'); *org-for-index*, where the organization name is used for an index that indicates its value ('*BMW slipped 4p to 31p*'); *othermet*, where the metonymy does not fall into any of the prespecified patterns ('*funds [. . .] had been paid into Barclays Bank.*').

[2] Examples were taken from Markert and Nissim (2007).

The dataset also includes annotated examples for two class-independent categories: *object-for-name*, where a name can be used as a mere signifier, not referring to any object or set of objects ('*Chevrolet is feminine because of its sound - it's a longer word than Ford, has an open vowel at the end, connotes Frenchness.*'); *object-for-representation*, where a name can refer to a representation (such as a photo or painting) of the referent of its literal reading ('*Look at the picture: this is Malta.*').

The occurrences of all names were extracted from the BNC. The total number of examples for *location* were divided into a training set of 925 and a test set of 908 annotated cases. The examples for *organization*, instead, were divided into a training set containing 1090 and a test set containing 842 annotated cases. The distribution of the literal, metonymic and mixed classes are 80%, 18% and 2%. The task was to automatically classify country and company names as having a literal or non-literal meaning given a four-sentence context. Participants in the SemEval 2007 Shared Task 8 could additionally attempt finer-grained classifications according to the prespecified set of metonymic patterns.

3.2.4. ReLocaR

One of the issues of the SemEval dataset is that it is too biased towards the literal class. In addition, the annotation strategies were challenged by Poibeau (2007), by Zhang and Gelernter (2015) and by Gritta et al. (2017). To address these issues, Gritta et al. (2017) introduced a new metonymy resolution (MR) dataset called ReLocaR (Real Location Retrieval), designed to evaluate the ability of models to correctly classify literal, metonymic and mixed location mentions.

It contains 1026 training and 1000 test examples collected using Wikipedia's Random Article API1[3]. Approximately 80% of sampled examples were literal, so the excess literal instances were discarded to balance the classes.

ReLocaR too has three classes: literal, metonymic and mixed. The first describes territorial interpretations, that is inanimate places that correspond to a set of coordinates ('*The treaty was signed in Italy.*'[4]). The second covers the occurrences of locations that express animacy ('*Jamaica's indifference will not improve the negotiations.*'), stand for any persons or organisations associated with it ('*We will give aid to Afghanistan.*'), a product ('*I really enjoyed that delicious Bordeaux.*'), a sports team ('*India beat Pakistan in the playoffs.*'), a governmental or other legal entity ('*Zambia passed a new justice law today.*'),

[3] https://www.mediawiki.org/wiki/API:Random
[4] Examples were taken from Gritta et al. (2017).

or an event (‘*Vietnam was a bad experience for me.*’). Finally, the mixed class is assigned either in the cases in which both readings are evoked at the same time (‘*The Central European country of Slovakia recently joined the EU*’), or in the cases in which there is not enough context to discriminate an interpretation from another possible one (‘*We marvelled at the art of ancient Mexico.*’).

3.2.5. WiMCor

Containing about 2000 samples or under, SemEval and ReLocaR are fairly small datasets and are inadequate for large-scale machine learning and statistical analysis. In order to make up for the insufficient coverage of the different ways in which metonymy can be observed in real-world data and for the low granularity of the annotation scheme, Mathews and Strube (2020) created a new corpus called WiMCor (Wikipedia Metonymy Corpus).

Samples pertaining to location names were collected using the English Wikipedia and DBpedia[5]. Wikipedia disambiguation pages were used to identify instances of metonymy, and DBpedia was used to check the category of the entity. The dataset was constructed in a semi-automatic way: metonymic pairs were generated as pairs of Wikipedia articles that are referred to by the same

[5]Dataset available here: https://www.dbpedia.org/

title but refer to different concepts. A key condition for this selection is that the two entities have a strong relationship. For example, the capital Paris and Paris Hilton are referred to by the same word 'Paris' but there is no strong link between them, so the instances do not form a metonymic couple. On the contrary, Delft and Delft University of Technology constitute a metonymic pair because the university is located in the city it gets its name from and both concepts can be referred to by the same anchor text. Samples are then generated automatically from Wikipedia articles. The entity occurring in that paragraph (restricted to be between 10 and 512 words long and generally composed of more than one sentence) is substituted with the anchor text, which corresponds to the PMW. For example, '*The Delft University of Technology applied for a patent […]*' is transformed into '*Delft applied for a patent […]*'[6].

The annotation scheme is built on various levels of granularity. The coarse-grained labels are *literal* and *metonymic*, identifying whether the location name refers to a geographical entity or has other possible interpretations. The medium-grained labels identify the type of the entity the PMW refers to, which can be *location*, *institution*, a*rtifact*, *team* and e*vent*. The fine-grained level of annotation specifies the precise entity the PMW refers to so, in the

[6] Examples were taken from Mathews and Strube (2020).

abovementioned case of Delft, we would have *Delft, Netherlands* or *Delft University of Technology*.

Only the pairs for which at least 50 samples were generated were retained. The final dataset contains 206.000 samples for 5404 metonymic pairs. The corpus is partitioned into train, validation and test set in the ratio 60:20:20 respectively.

3.3. A more representative dataset

All the datasets introduced so far are concerned with location names and how they can be used metonymically to refer to people, organizations, events or products. This is, however, just one of the many classes that can dynamically acquire new meaning in metonymy (see the rich taxonomy by Radden and Kövecses (1999) in Section 3.1.).

A new dataset of 509 items was created that is representative of the most common types of metonymies (Pedinotti and Lenci, 2020). The relationships included are container for content, producer for product, product for producer, location for located, causer for result and possessed for possessor. Each item consists of two sentences, one in which a PMW is used metonymically and one in which it is used literally, both accompanied by a paraphrase making the meaning of the target word explicit. For example, if the target word is '*bottle*',

there will be a sentence in which the target word is used metonymically (e.g., '*The guest tasted the bottle*'), the metonymic paraphrase (e.g., '*wine*'), a sentence in which the target is used literally (e.g., '*The man raised the bottle*') and the literal paraphrase (e.g., '*container*')[7].

The dataset was used to test whether deep neural architectures for language modelling and in particular BERT (Devlin et al., 2019) contextualized embeddings can be used to model the meaning shifts associated with metonymic uses of words.

[7] Example taken from Pedinotti and Lenci (2020).

4. The creation of a new dataset

In this section, the new dataset is presented. This includes the description of the different steps of construction of our collection of data: the types of metonymy included (Section 4.1.), the collection and annotation of metonymic sentences (Section 4.2.), the validation of metonymic sentences (Section 4.3.) and the collection and annotation of negative examples (Section 4.4.). Finally, some statistics from the complete dataset are shown (Section 4.5.). The purpose of this work is to create a dataset for testing pre-trained language models representations. The goal is to find whether such representations are able to detect and encode meaning shifts in sentences. To do so, we present a dataset containing a higher number of metonymic relationships than the ones introduced so far for a more complete probing of the models with respect to metonymy classification.

4.1. Types of metonymy included in the dataset

Section 3.2. contains a brief survey of the existing datasets for metonymy. We have seen how the datasets used so far consists mainly of metonymies of the

location type such as '*England lost in the semi-final*'[8]. The aim of the new dataset is to collect examples for different kinds of metonymic relationships to create a dataset that is more representative of the use of metonymy in natural language.

Sentences were collected for the following metonymic relationships:

1. Contingency
2. Event-agent
3. External Component
4. Internal Component
5. Origin
6. Participant
7. Patient
8. Attributive
9. Productive
10. Spatial
11. Temporal

Some examples of metonymic pairs and sentences for each relationship are shown in Table 1.

[8] Example from Nissim and Markert (2003).

Relationship	**Example metonymic pair**	**Sentence**
Contingency	bus - fare	*Passengers should pay the bus when boarding.*
Event-agent	doctor-appointment	*Do not skip the doctor.*
External Component	sail - boat	*Their 36-foot sail hit rough waters, losing power and the ability to steer.*
Internal Component	pen - ink	*He also walked around in public with pen smudged on his face.*
Origin	author - book	*He also supported the publication of other authors.*
Participant	ambulance - people	*The ambulance used a defibrillator on the patient and assisted his breathing.*

Patient	airplane - pilot	*Eight seconds later, the airplane acknowledged that and said he had two helicopters in sight.*
Attributive	beauty - woman	*This Sally Keeble, an absolute young beauty, could not be fooled.*
Productive	cigarette – smoke	*Nicotine affects the brain within seconds of inhaling a cigarette.*
Spatial	beer - bottle	*Canon is accused of trying to cut a relative with a broken beer during an argument at her home.*
Temporal	Hiroshima - bomb	*This was not ended till August 8, two days after Hiroshima, when the Soviets declared war.*

Table 1. Examples of word pairs and sentences for each metonymic relationship in the dataset.

A complete description of the metonymic word pairs included in the dataset is given in Section 4.5, together with an illustration of the distribution of data in the resulting dataset. Metonymic and negative examples (sentences in which the

target word is used with a literal interpretation) for each pair can be seen in Appendix A.

4.2. Data collection and annotation

In the first phase, for each of the relationships listed above, a number of sentences was extracted automatically from ukWac (Baroni et al., 2019), BNC (BNC Consortium, 2007) and the Wikipedia Corpus[9]. We had a relevant number of metonymic word pairs for each relationship, each having dozens of examples. These were annotated by two annotators: me and Paolo Pedinotti, a PhD student. Only the sentences we both agreed on were included in the first collection of data. The data included the kind of relationship, the cue word (CW), the target word (TW), the original sentence in which the TW is used with its literal interpretation, the metonymic sentence in which the TW is used metonymically, and the index of the TW and CW in the original sentence and of the TW in the metonymic sentence. Here is an example (Table 2):

[9] Designed by Mark Davis in 2015, it is accessible at https://www.english-corpora.org/wiki/

relation	Spatial
cue word	beer
target word	bottle
original sentence	*A grocery store sells unique and hard to find* beers in bottles, *which you can either drink on premise or take home.*
metonymic sentence	*A grocery store sells unique and hard to find* bottles, *which you can either drink on premise or take home.*
id cue word	10
id target word	12
id metonymy	10

Table 2: The information included in the first collection of data.

The metonymic sentences were generated automatically starting from the original sentences, so they sometimes needed editing or cleaning to be grammatically or idiomatically complete. This was done in a different column and the indexes were corrected accordingly. The resulting collection was not large enough. In fact, we had reached about 200 sentences, but the aim was to collect 1000 metonymic sentences to present to and validate with native speakers (Section 4.3.). I manually collected the remaining sentences on Sketch

Engine[10], an online tool that allows to interrogate corpora, and added the missing information. To do so I used specific queries:

- *[ws("beer-n", "\"%w\" [^ /]+ \.\.\.", "bottle-n")]* to find sentences containing prepositional phrases such as *beer in bottle*, *beer in the bottle*, *beer of the bottle*, etc.
- *[ws("bottle-n", "modifiers of \"%w\"", "beer-n")]* to find sentences where the TW modifies the CW such as *bottle beer*.
- *[ws("beer-n", "possessors of \"%w\"", "bottle-n")]* to find sentences containing the genitive form such as *bottle's beer*.

The sentences retrieved were manually edited to obtain the metonymic form. For example, '*I recall drinking mouthfuls of increasingly warm beer from random bottles which others had left behind'* was edited as *'I recall drinking mouthfuls of increasingly warm random bottles which others had left behind'*. These were again approved by the second annotator. All sentences included in the collection have a maximum length of 30 tokens (both words and punctuation are counted). Once 1000 sentences were collected, we moved on to the next step concerning their validation with native speakers.

[10] https://www.sketchengine.eu/

4.3. Data validation

The collected data was validated through Prolific[11], a platform that allows to launch studies to thousands of participants that can be filtered and selected according to some criteria. The study was presented as a *study on the acceptability of sentences* and the task was defined as follows:

'How acceptable is each sentence?

Please rate the acceptability of the following sentences on a seven-point scale. A score equal to 1 must be selected if the sentence makes no sense and its use is not plausible (e.g. 'Perhaps I just need to get used to driving a cigarette'); a score equal to 7 means that the sentence is perfectly plausible (e.g. 'Airplane pilots have long used checklists before take off to ensure safety')'.

8 questionnaires were submitted to 5 participants each. These were filtered according to their native language and their reliability in past studies. Only native English speakers who had participated in and whose results had been accepted in past studies were selected.

Each questionnaire included 50 sentences, 10 of which were control examples: 5 were meaningless sentences for which we expected a score equal or close to

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

1, and the other 5 were perfectly acceptable sentences for which we expected a score equal or close to 7. The meaningless sentences were created by taking some sentences in the collection and substituting one word with another one. These are:

1. *Margaret More composed milk, which she played on stage.*
2. *Four Latino men sat at the bar drinking guitar.*
3. *It is approximately eight times warmer than a pen and does not felt or shrink.*
4. *She hated ballet; the mug made her feet bleed.*
5. *This morning, she poured herself a cottage and examined the raw goat milk.*

The meaningful sentences were either examples of highly conventionalized metonymies or sentences containing no metonymy at all:

1. *Nixon bombed them anyway.*
2. *According to the State Patrol, Smith was driving a Ford.*
3. *His words will be accompanied by the traditional music of the Celtic harp.*
4. *We can go back now and ask about the health effects following the Hiroshima bombs.*

5. *Most of those have to go upstairs and I'll need to hire some muscle to do that.*

In order to judge participants as reliable and therefore accept their results, the average score assigned to the meaningless sentences had to be smaller than 3, while the average score assigned to the meaningful sentences had to be higher than 5. The remaining 40 sentences were heterogeneous examples from the collection introduced in Section 4.2.

To obtain a robust annotation, mean and variance were measured on the total judgements for each sentence. If the mean was equal to or greater than 4.5 (meaning that participants find the sentence acceptable) or equal to or smaller than 2 (meaning that participants find the sentence not acceptable), and the variance was equal to or smaller than 3.5, the annotation was considered robust. Otherwise, the sentences were collected and tested a second time in the ninth final questionnaire.

For the metonymic pairs for which the examples shown to participants were always robustly rated as acceptable, all sentences in the collection were included in the final dataset. For CW-TW pairs that were sometimes rated as acceptable, only the examples shown to participants that obtained a high score were selected. Finally, for the CW-TW pairs always rated as not acceptable, no example was taken. This is the case for only one pair, *beard* for *man* (‘*Next to*

her is a large gray beard who is drinking a frothy beer'). The distribution of data is shown in Section 4.5.

4.4. Collection and annotation of negative examples

The collection of negative examples, that is to say sentences not presenting metonymy at all, was straightforward. The sentences in which the TW is used in its literal interpretation were selected from a collection of automatically retrieved sentences. These were again annotated by me and Dott. Paolo Pedinotti and only the ones we both agreed on were selected. The ratio is 3:1, meaning that for each metonymic sentence, 3 sentences in which the word is used literally were collected.

4.5. Dataset statistics

In this section the distribution of data in the final dataset is shown according to relation and metonymic pairs. The richness in terms of variety of relationships between the cue and the target words is the main novelty of this dataset with respect to existing datasets described in Section 3.2. 504 sentences are

metonymic, 1512 are literal. The total number of CW-TW pairs is 42. The details are illustrated in the table below (Table 3).

Relation	**CW-TW pair**	**N. of metonymic examples**	**N. of literal examples**
Contingency	fare bus	27	81
Event-agent	appointment doctor	6	18
	birth baby	22	66
External Component	boat sail	6	18
	car door	3	9
	car wheel	4	12
	door barn	1	3

	guitar string	20	60
	wool sheep	4	12
Internal Component	ink pen	6	18
	milk coconut	5	15
Origin	book author	6	18
	author book	39	117
Participant	people ambulance	50	150
	people boat	8	24
	people building	5	15
Patient	battery car	4	12

	car battery	2	6
	people motorcycle	6	18
	pilot airplane	24	72
	pilot helicopter	5	15
	shoe toe	6	18
Attributive	diaper baby	13	39
	officer uniform	4	12
	woman beauty	27	81
Productive	chord guitar	2	6
	music guitar	2	6

	music harp	5	15
	music piano	5	15
	music violin	23	69
	smoke cigar	23	69
	smoke cigarette	19	57
	sound keyboard	21	63
Spatial	beer bottle	3	9
	bottle beer	33	99
	carton milk	4	12
	coffee mug	4	12

	game ball	7	21
	milk carton	19	57
	mug coffee	6	18
Temporal	bomb Hiroshima	23	69
	vacation cottage	2	6

Table 3: Distribution of data in the final dataset.

5. Experiments

In this section the experiments carried out on the new dataset are discussed. The aim of this work is to probe pre-trained language models for metonymy classification. The idea is that we can test the sentence embeddings created by these large models by using a simple classifier to detect whether the information we are interested in is encoded in such representations and is easily accessible (see Section 2.2. for more details on probing tasks). The pre-trained model used for generating sentence embeddings for our dataset is BERT (Devlin et al., 2019). This is done with and without fine-tuning the model on the dataset. The representations obtained are then passed to the classifier, the performance of which is an indicator of the availability and accessibility of information concerning meaning shift in the given embeddings. The task carried out by the probe is indeed a binary classification: a sentence can either contain a metonymy or not.

First, I introduce the model used in Section 5.1., with regard to the pre-trained language model (Section 5.1.1.) and the probe (Section 5.1.2.). Then, I describe the different test sets on which the experiments were run (Section 5.2.), the classification task (Section 5.3.) and the corresponding performances (Section 5.4.). Finally, I analyse the results in more detail in Section 5.5.

5.1 Models

The models introduced in this section are the pre-trained language model (Section 5.1.1.) used to create the embeddings for the sentences in the new dataset, and the probe (Section 5.1.2) trained to perform a binary classification task on such embeddings. The model assessed is BERT (Devlin et al., 2019), a deep architecture pre-trained on large amounts of linguistic data that currently represents the SoTA in NLP on many language tasks. The probe is a much simpler model selected specially for this task.

5.1.1. BERT

BERT (Bidirectional Encoder Representations from Transformers; Devlin et al., 2019) is a language representation model designed to train bidirectional deep representations. Unlike unidirectional language models like OpenAI GPT (Radford et al., 2018), where every token can only attend to previous tokens in the self-attention layer of the Transformer (Vaswani et al., 2017), BERT's representations are trained by jointly conditioning on both left and right contexts in all layers.

The architecture of BERT is a multi-layer bidirectional Transformer encoder. The number of layers (i.e., Transformer blocks), the hidden size and the number

of self-attention (for more details on Attention see Vaswani et al., 2017, and Alammar, 2018c) heads define two model sizes: BERT-base and BERT-large. BERT-base has 12 layers, a hidden size of 768, 12 self-attention heads and 110M parameters in total. BERT-large, instead, has 24 layers, hidden size 1024, 16 self-attention heads and 340M total parameters.

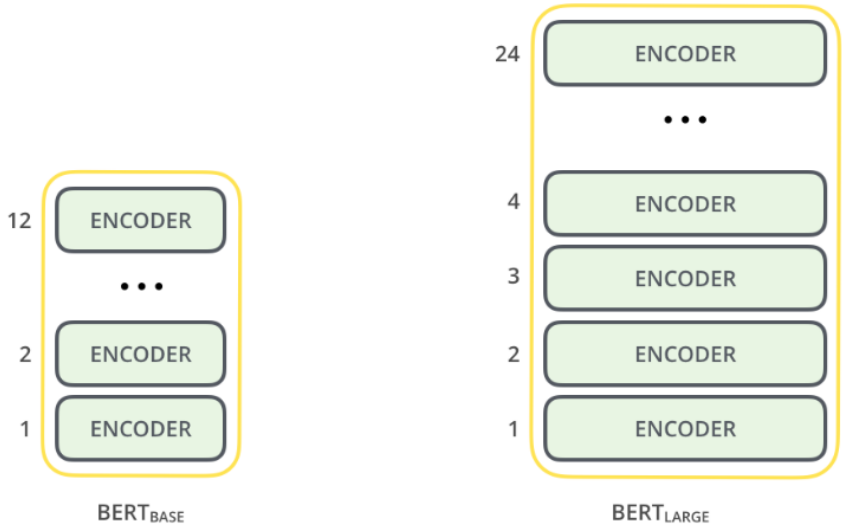

Figure 2: BERT base and BERT large (Image from Alammar, 2018b).

The input representation is able to encode a sentence or a pair of sentences in one token sequence. This allows BERT to handle a variety of downstream tasks.

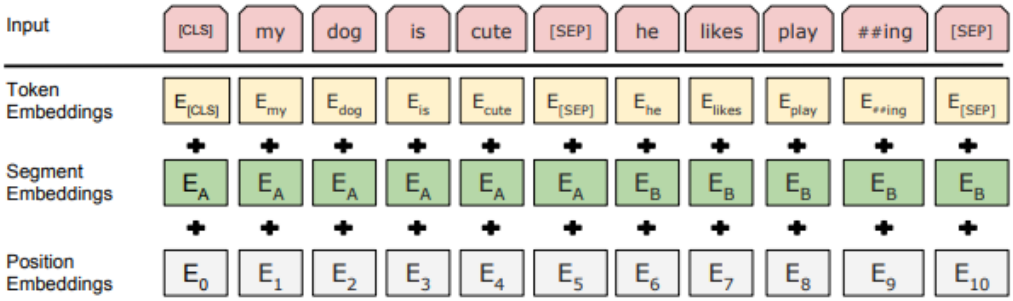

Figure 3: BERT input representation (Image from Devlin et al., 2019).

A special classification token (*[CLS]*) is always the first token of every sequence. The aggregate sequence representation for classification tasks is the final hidden state corresponding to this token. Sentence pairs are concatenated in a single sequence and separated by the special token *[SEP]*.

BERT's input is a sequence of words. Each layer applies self-attention to the sequence, passes the result through a feed-forward network and then to the next encoder.

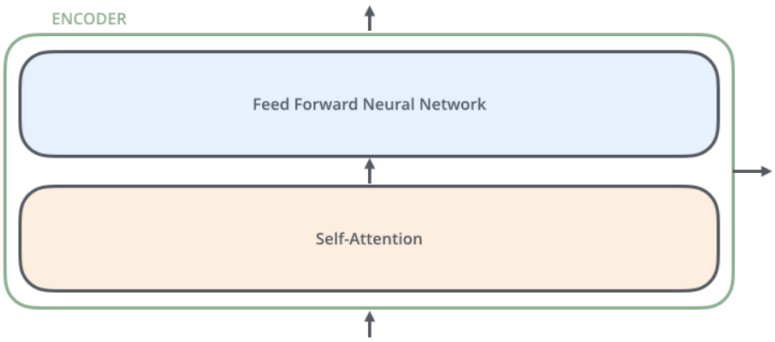

Figure 4: Two sub-layers of an encoder. Encoders are identical in structure but do not share the same weights (Image from Alammar, 2018a)

For each position the output is a vector of size 768 in BERT-base. For sequence classification, only the output of the first position (to which the special token [CLS] was assigned) is considered.

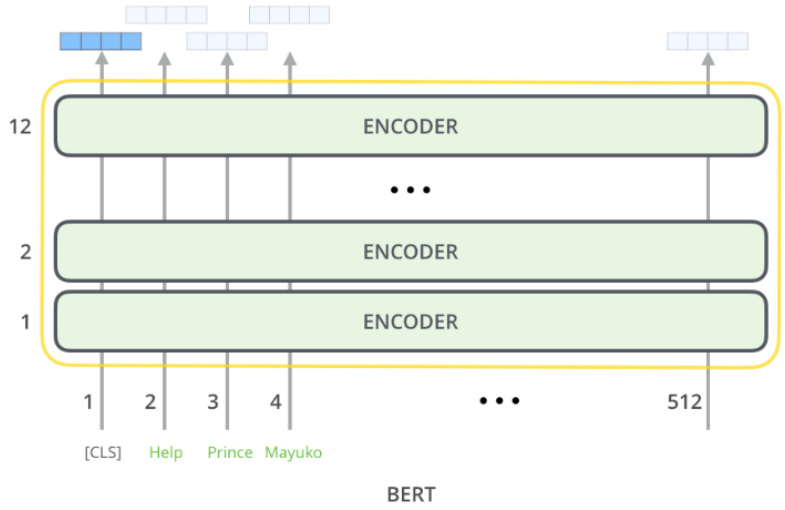

Figure 5: Model output (Image from Alammar, 2018b).

The output vector of the first position can be used as the input for a classifier.

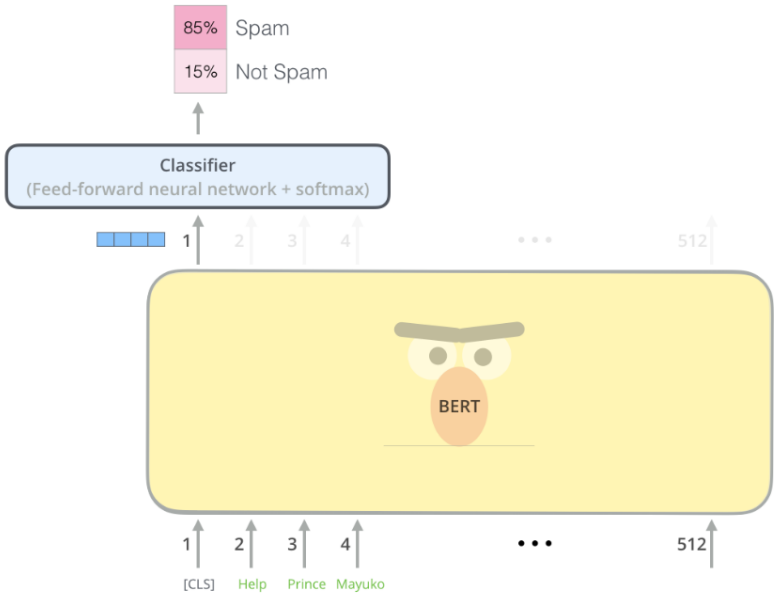

Figure 6: Example of classification task with BERT's output (Image from Alammar, 2018b).

There are two steps to the impementation of BERT: pre-training and fine-tuning. During the first step the model is trained on unlabelled data on different pre-training tasks. To fine-tune the model, it needs to be initialized with the pre-trained parameters which are then fine-tuned using labelled data from the downstream task.

To enable pre-trained deep bidirectional representations, BERT uses a masked language model (MLM) and a binarized next sentence prediction (NSP) task. In the MLM, some percentage of the input tokens are masked randomly and

predicted. This procedure is also referred to as a *cloze task* in literature. In the NSP task, two sentences A and B are given: in 50% of cases B is the sentences that actually follows A, in the other 50% of cases B is a random sentence. This allows to capture the relationship between two sentences, an information needed for many downstream tasks such as Question Answering (QA) and Natural Language Inference (NLI). BERT is pre-trained on the BooksCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2,500M words).

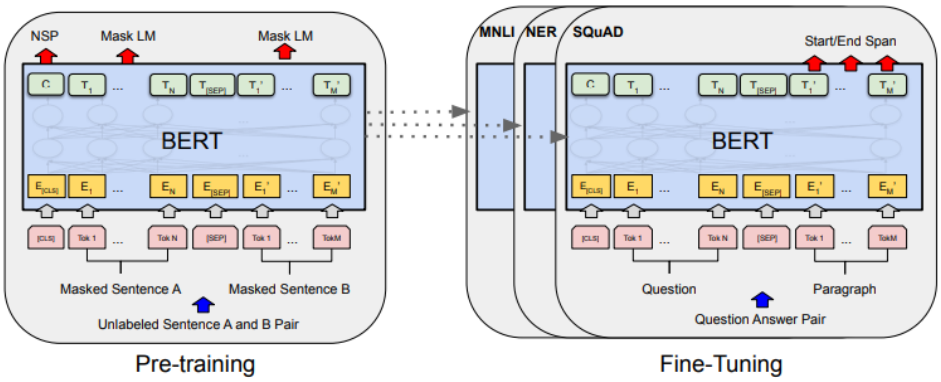

Figure 7: Pre-training and fine-tuning processes for BERT. The architecture remains the same, but all parameters are fine-tuned during fine-tuning (Image from Devlin et al., 2019).

Fine-tuning is relatively inexpensive compared to pre-training. It requires an additional output layer and training on a dataset annotated for a specific task, which results in a modification of the network weights via back-propagation of

the error. Therefore, the pre-trained model can be fine-tuned for a wide range of tasks without substantial task-specific architecture modifications.

For the classification task presented in Section 5.3. I used the sentence representations created by BERT-base-uncased[12] with and without fine-tuning on the new dataset. The model lowercases words starting with a capital letter. For the fine-tuning of the model, I used BertForSequenceClassification[13] pre-trained model.

This is the summary of the architecture of one of its 12 hidden layers:

```
BertForSequenceClassification(
  (bert): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0): BertLayer(
          (attention): BertAttention(
            (self): BertSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
```

[12] https://huggingface.co/tftransformers/bert-base-uncased
[13] https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertForSequenceClassification

```
            (key): Linear(in_features=768, out_features=768, bias=True)
            (value): Linear(in_features=768, out_features=768, bias=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
          (output): BertSelfOutput(
            (dense): Linear(in_features=768, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
        (intermediate): BertIntermediate(
          (dense): Linear(in_features=768, out_features=3072, bias=True)
          (intermediate_act_fn): GELUActivation()
        )
        (output): BertOutput(
          (dense): Linear(in_features=3072, out_features=768, bias=True)
          (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
```

This model offers an additional argument to add an optional classification head with the required number of labels.

```
model = BertForSequenceClassification.from_pretrained("bert-base-uncased", num_labels=2)
```

This adds a sequence classification head with two output units as the final layer (a linear layer on top of the pooled output):

```
)
    )
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (dropout): Dropout(p=0.1, inplace=False)
  (classifier): Linear(in_features=768, out_features=2, bias=True)
)
```

The parameters used for the fine-tuning process are:

Learning rate (Adam): 2e-5;

Batch size: 32;

Epochs: 3;

Max_sequence_length: 30;

Dropout: 0.1.

While most parameters follow the ones proposed for fine-tuning by Devlin et al. (2018), the maximum sequence length, which controls the length of padding and truncation, was set to 30 after the distribution of the sentence length in the dataset, which is between 5 and 30.

5.1.2. Probing classifier

In order to identify the best probing classifier for the task, 3 potential probes of increasing complexity were tested, compared to a baseline classifier and evaluated with a control task.

The first model is a linear perceptron classifier imported from scikit-learn[14]:

```
perceptron = Perceptron(validation_fraction=0.2, random_state=42, penalty='elasticnet')
```

The second is a multi-layer perceptron classifier with one hidden layer of 10 nodes imported from the same library[15] and implemented with the configuration of parameters that allowed the best performance:

```
MLPclass = MLPClassifier(hidden_layer_sizes=(10,), max_iter=150, activation='relu', solver='lbfgs', verbose=1, random_state=42, learning_rate='invscaling', validation_fraction=0.2)
```

[14] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html
[15] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

The last probe was built in Keras[16]. It is a sequential model with 3 layers defined as follows:

```
model = Sequential()

model.add(Dense(20, activation='relu'))

model.add(Dropout(0.4))

model.add(Dense(15, activation='relu'))

model.add(Dropout(0.4))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',                optimizer='adam',
metrics=['accuracy', f1_m, precision_m, recall_m])
```

First, I compared the performances of the three models against the performance of a baseline. The idea is to observe whether the probes perform significantly better than the baseline. I used a classifier that makes predictions ignoring the input features. A DummyClassifier was imported from scikit-learn[17] which generates predictions uniformly at random from the list of unique labels, meaning that each class has equal probability.

[16] https://keras.io/
[17] https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html

The probing classifiers were tested on the classification task described in Section 5.3. Specifically, the sentence embeddings generated by BERT-base-uncased without fine-tuning were given as input. The test set, made up from 30% of examples in the dataset, was defined through random split. The classifier's task was to assign the correct label to each sequence: 'Literal' or 'Metonymic'. The F1-score for each model is shown in Table 4.

Classifier	**Performance (F1 score)**
Perceptron	0.47
MLPClassifier	0.58
KerasSequential	0.52
DummyUniform	0.34

Table 4: F1 scores of probing classifiers on binary classification task.

As all classifiers outperform the baseline, therefore their linguistic task accuracy is acceptable, they are evaluated on a control task to find whether they also exhibit high selectivity (Hewitt and Liang, 2019). Hewitt and Liang (2019) define control tasks which associate the inputs to random outputs. The idea is that this information can only be learnt by the probe itself, so if the probe performs well on the control task it means that is has low selectivity. This is due to the fact that the more a probe's ability to make output decisions is independent

from the linguistic properties of a representation, the less its accuracy on a linguistic task is descriptive of the properties of a representation. The goal is to identify a probe that has high accuracy on the linguistic task and low control accuracy, meaning that it is actually learning from the representations given in input rather than memorizing and performing similarly on any data.

The control task for this work was defined by randomly assigning the two possible labels to the sentences while conserving the original ratio. Selectivity was measured as the difference between the F1 score of each classifier on the control task and the actual classification task. The results are shown in Table 5.

Classifier	**Class. Task**	**Control Task**	**Selectivity**
Perceptron	0.47	0.14	0.33
MLPClassifier	0.58	0.22	**0.36**
KerasSequential	0.52	0.18	0.34
DummyUniform	0.34	0.34	0.00

Table 5: Selectivity of probes as the difference between the F1 scores on control task and real classification task.

While the Dummy Classifier performs equally on both tasks, therefore obtains a selectivity of 0, Perceptron, MLP Classifier and Keras Sequential present a similar selectivity, respectively of 0.33, 0.36 and 0.34. The output decisions that

display the higher dependence on the linguistic properties of the representations are the ones made by MLP Classifier, with an F1 score equal to 0.22 on the control task and to 0.58 on the linguistic classification task. As a consequence, the probing classifier used from this point on for the experiments presented in the thesis is MLP Classifier.

5.2. Train/test split

Experiments were carried out in different settings according to the intended test set. The train-test split was controlled to verify the generalization level for the linguistic information. There are two different train and test sets for each experiment.

The first is a random split in which the test set is made up of 30% of the examples in the dataset. The size of the training and test sets are respectively 1411 and 605. The distribution of sentences according to their label is shown in Table 6.

Training set		**Test set**	
Literal	**Metonymic**	**Literal**	**Metonymic**
1050	361	462	143

Table 6: Distribution of examples in training and test set for random split.

The number of examples for every metonymic relationship included in the test set is (Table 7):

Metonymic relationship	**N. of examples in test set**
Contingency	4
Event-agent	10
External Component	14
Internal Component	3
Origin	16
Participant	24
Patient	10
Attributive	15
Productive	24
Spatial	19
Temporal	4

Table 7: Number of metonymic examples for each relationship in the test set obtained through random split.

The second train/test split was designed to include in the test set only examples of TW-CW pairs that are not present in the training set, as in Ettinger at al. (2018). This is done to verify whether the model is able to generalize the

linguistic information learnt from representations and to apply this knowledge to examples never seen in the training phase.

To do so, I alphabetically ordered the word pairs and split the dataset in a way that would maintain a similar training and test set proportion with respect to the random split described above. All sentences for the last 15 pairs make up the test set. The size of the training and test sets are respectively 1452 and 564. The distribution of sentences according to their label is shown in Table 8.

Training set		**Test set**	
Literal	**Metonymic**	**Literal**	**Metonymic**
1089	363	423	141

Table 8: Distribution of examples in training and test set for controlled split.

The metonymic relationships the word pairs represent are (Table 9):

Metonymic relationship	**N. examples in test set**
External Component	34
Internal Component	6
Patient	17
Attributive	4
Productive	49

Spatial	8
Temporal	23

Table 9: Number of metonymic examples for each relationship in the test set obtained through controlled split.

5.3. Classification task

The aim of the experiments here presented is to interpret language models' representations and verify the linguistic properties they encode. A classification task is designed for probing specific information captured in vector representations of sentence meaning. This approach builds on diagnostic methods proposed by Gupta et al. (2015) and Ettinger et al. (2016), which involve a classification task for targeting some information captured in vector representations. The idea is that if we have a collection of composed vectors that represent sentences and we require a classifier to identify a particular type of semantic information, by measuring the quality of the performance of the classifier on this task we can assess whether the information in question is present in the composed representations and is accessible (Ettinger et al., 2016).

We have constructed and annotated a dataset for metonymy (Section 4.) in which some target words are either used in their metonymic or literal interpretation. Once the pre-trained language model (Section 5.1.1.) and the

probing classifier (Section 5.1.2.) have been selected, the classification task can be carried out. The vector representations generated by the language model are given as input to the probe, whose task is to label each sequence as 'Literal' or 'Metonymic'. This is done in different combined conditions: we consider a) the representations created by BERT (Devlin et al., 2019) with and without fine-tuning on the new dataset and b) the different training/test splits described in Section 5.2. The goal is to ascertain whether pre-trained language models are able to encode information relating to the semantic shift produced by metonymy.

5.4. Performances

This section reports the performances of the probe on the classification task taking into consideration the sentence embeddings created by BERT (Devlin et al., 2019) with and without fine-tuning and the different test sets.

The results of the classifier for the data setting in which the training and the test sets are split randomly are shown in Table 10:

BERT with no fine-tuning	**BERT with fine-tuning**
F1 score: **0.58**	F1 score: **0.52**
Precision: 0.56	Precision: 0.54
Recall: 0.60	Recall: 0.50
Confusion matrix:	*Confusion matrix:*
397 65	401 61
57 86	71 72

Table 10: Performances of the probe with the random train/test split.

The results for the second data setting in which the train/test split was controlled in order to only include in the test set examples never seen during the training phase are (Table 11):

BERT with no fine-tuning	**BERT with fine-tuning**
F1 score: **0.48**	F1 score: **0.45**
Precision: 0.57	Precision: 0.56
Recall: 0.41	Recall: 0.38
Confusion matrix:	*Confusion matrix:*
380 43	382 41
82 59	87 54

Table 11: Performances of the probe with the controlled train/test split

F1 scores are always higher when the tested sentence embeddings are obtained from the model with no fine-tuning. This might be due to the small number of examples in the new dataset which does not allow for a robust generalization on the phenomenon. At the same way, when moving from a random to a controlled train/test split, F1 scores decrease of almost 0.10. This is unsurprising as the probe has never seen the examples it is tested on. The scores are, however, still higher than both the baseline introduced in Section 5.1.2. (*DummyUniform*, F1 score: 0.34*)* that performs equally in each setting, and the F1 score obtained by the probe on the control task (F1 score: 0.22). This means that the embeddings do carry some information encoding metonymy, but it is either fuzzy, incomplete or difficult to retrieve or interpret.

5.5. Results analysis

This section includes a deeper look into the probing task outcome. By analysing the classifier's predictions, it is possible to study F1 scores for the different classes of metonymic examples in the test set. The following results consider only the metonymic sentences in the test set, so are measured on how many metonymic examples were predicted correctly. As seen in Section 5.2., the test set obtained through random split includes examples for all of the 11 target and

cue words relationships, while the controlled test set includes examples for 7 metonymic classes.

In Table 12, F1 scores for each metonymic relationship in each of the 4 different settings are shown. Most of the highest scores are given by non fine-tuned representations and a random test set: examples for *event-agent* (0.66, e.g. '*Obviously any* baby *is a joyous event'*), *external component* (0.66, e.g. '*The club currently has two* sails *docked at the Newport Yacht Basin*'), *internal component* (0.66, e.g. '*The ground rice is then soaked in* coconut *until it is soft'*), *origin* (0.70, e.g. '*Do you read other* authors*?*') and *productive* (0.71, e.g. '*He was dead drunk and blew his* cigar *at my face several times*') achieve a score greater than 0.65. The same happens for *internal component* (0.66) in the controlled split setting for both fine-tuned and non fine-tuned representations, for *patient* (0.68, e.g. '*When waiting at a stoplight, why do* motorcycles *rev their throttle?*') on the controlled split test set for non fine-tuned representations and for *participant* (0.67, e.g. '*The* ambulance *used a defibrillator on the patient and assisted his breathing*') on fine-tuned representations and controlled train/test split.

CW-TW relation	**No fine-tuning, random split**	**No fine-tuning, controlled split**	**Fine-tuning, random split**	**Fine-tuning, controlled split**
Contingency	0.25	-	0.27	-
Event-agent	**0.66**	-	0.56	-
External Component	**0.66**	0.30	0.52	0.29
Internal Component	**0.66**	**0.66**	0.36	**0.66**
Origin	**0.70**	-	0.60	-
Participant	0.60	-	**0.67**	-
Patient	0.45	**0.68**	0.35	0.47
Attributive	0.47	0.28	0.44	0.57
Productive	**0.71**	0.37	0.42	0.50
Spatial	0.61	0.45	0.45	0.50
Temporal	0.50	0.20	0.36	0.20
Average F1 score per experiment setting	**0.57**	**0.42**	**0.46**	**0.46**

Table 12: F1 scores for each metonymic class for not fine-tuned and fine-tuned representations and for different test sets.

The results analysis also displays some low scores, particularly on *contingency* (0.25, 0.27, e.g. '*This ticket will include your* bus'), where the result improves with the fine-tuning of the pre-trained model, *external component* (0.30, 0.29) on the controlled test set, and *temporal* (0.20, 0.36, 0.20, e.g. 'Hiroshima *killed thousands of children*') in most settings.

It is interesting to note how, in general, test examples in the controlled split only achieve lower scores when the representations are not fine-tuned, otherwise their average score is equal to the one obtained by fine-tuned representations in the random split condition. This may suggest that, even though on average the fine-tuning of the model slightly decreases scores, it provides for some sort of robustness that guarantees coherency in the recognition of metonymy even for examples the classifier has never seen before.

Indeed, we are more interested in how the classifier performs on the controlled test set, as its ability to properly classify new examples is an indicator of higher-quality representations with respect to the linguistic phenomenon. Table 13 focuses specifically on the classifier's results on the controlled test set, and shows how in most cases, 5 out of 7, the score remains equal or increases with the fine-tuning of the model.

CW-TW relation	**No fine-tuning, controlled split**	**Fine-tuning, controlled split**
External Component	0.30	0.29
Internal Component	0.66	**0.66**
Patient	0.68	0.47
Attributive	0.28	**0.57**
Productive	0.37	**0.50**
Spatial	0.45	**0.50**
Temporal	0.20	**0.20**
Average F1 score per experiment setting	0.42	0.46

Table 13: Focus on scores for controlled test set.

This means that, even though at a first sight the sentence embeddings created with pre-trained BERT with no fine-tuning seem to be the best ones with respect to the linguistic information encoded, fine-tuning the model actually does improve the representations by adding some information that allows the probing classifier to generalize to some extent and find it in new metonymies.

Only in two cases the score decreases for fine-tuned representations: it is the case of *external component*, where the variation is minimal (from 0.30 to 0.29) and *patient* (from 0.68 to 0.47).

It is important to also bear in mind that the dataset is biased towards the 'literal' class, which could have negative effects on the training of the classifier, and it is fairly small, so does not present the optimal size for an efficient fine-tuning.

6. Conclusion

The objective of this thesis is to explore sentence representations obtained from pre-trained language models. Having attained remarkable accuracies, models such as BERT (Devlin et al., 2019) represent the SoTA for NLP but are difficult to interpret, meaning that what linguistic information these models are encoding is still in part undiscovered. The issue of low interpretability can be tackled through probing methodologies, which aim at identifying the presence of some specific information of interest in the vector representation. The idea as that a classifier can be trained on pre-trained embeddings and asked to classify the inputs according to some linguistic property; if the classifier succeeds in the task, we assume that the embedding captures the linguistic information being researched.

The linguistic phenomenon being studied in this work is metonymy, which occurs when an entity is referred to with the name of another entity to which it is closely related. In spite of the fact that the relationship between the two entities can be of different types, most existing datasets for the study of metonymy focus on the *location* one, where a location name is used to refer to an institution, an event or a product.

For the purposes of this work a new dataset for metonymy of 2016 sentences was created which comprises 42 word pairs representing 11 different metonymic relationships. The vector representation of each sentence was created with BERT (Devlin et al., 2019), a transformer-based pre-trained language model. Embeddings were extracted with and without fine-tuning the model, BERT-base-uncased. The probe used is a multi-layer perceptron, the classifier that displayed the highest selectivity. The train and the test set were split at random and in a controlled way: in the first case, 30% of sentences from the dataset were selected randomly to make up the test set; in the second, 30% of sentences from the dataset were selected so that the sentences in the test set are representative of metonymic word pairs that are not included in the train set: the idea is to test how robustly pre-trained language models capture information related to metonymy by verifying if such information is traceable even in examples the classifier has never encountered before.

On the controlled test set, the performances of the classifier show that the information encoded in the sentence embeddings allows for better generalization on new data when the model is fine-tuned. However, the average F1 score is 0.46. While still being a higher score than the ones achieved by both the baseline and by the probe trained on a control task, it

is not an excellent result. This might mean that some linguistic information related to metonymy is encoded in the sentence embeddings, but it is either hazy, noisy, incomplete or difficult to retrieve.

7. Future Work

The research on how metonymy is captured by pre-trained language models has still large areas available for exploration. The models, the input, the dataset and the test set can all be exploited in many and diverse ways.

Firstly, other SoTA pre-trained models could be probed for metonymy. The input could be manipulated to place emphasis on the metonymic word, for example by concatenating the target word or the potentially metonymic word to the sentence in which it occurs, or some information on its position.

The dataset allows for the greatest margin of improvement, especially on size and representation. Indeed, the number of sentences could be increased and the two classes balanced in order to reduce bias, respectively improving the quality of the fine-tuning of pre-trained models and training of the classifier. Furthermore, examples for a higher number of metonymic word pairs could be included to make the dataset even more representative of the distribution of metonymy in natural language.

In the current controlled test set the focus was placed on word pairs rather than relationships, so the test set is comprised of word pairs never seen in the training phase that were selected alphabetically, not according to the relationship they are representative of. Indeed, only 7 of the 11 metonymic relationships are

included in the current controlled split. Therefore, manipulating the training and test set so that the test set contains examples for metonymic word pairs for all relationships could also be an interesting experiment.

8. Bibliography

Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., & Goldberg, Y. (2016). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Alammar, J. (2018a). The illustrated transformer. *The Illustrated Transformer–Jay Alammar–Visualizing Machine Learning One Concept at a Time*, *27*.

Alammar, J. (2018b). The illustrated BERT, ELMo, and co. *How NLP Cracked Transfer Learning*, *38*.

Alammar, J. (2018c). Visualizing A neural machine translation model (mechanics of Seq2seq models with attention). *Visualizing Machine Learning One Concept at a Time Blog*.

Baroni, M., Bernardini, S., Ferraresi, A., & Zanchetta, E. (2009). The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, *43*(3), 209-226.

Belinkov, Y. (2022). Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics, 48*(1), 207-219.

Blacoe, W. & Lapata, M. (2012). A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Language Processing and Computational Natural Language Learning*, pp. 545–56. Stroudsburg, PA: Assoc. Comput. Linguist.

BNC Consortium. (2007). British national corpus. *Oxford Text Archive Core Collection.*

Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5: 135-146.

Boleda, G. (2020). *Distributional semantics and linguistic theory. Annual Review of Linguistics*, 6:213–23.

Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326.*

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877-1901.

Cer, D., Yang, Y., Kong, S. Y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y., Strope, B. & Kurzweil, R. (2018). Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Chakraborty, S., Tomsett, R., Raghavendra, R., Harborne, D., Alzantot, M., Cerutti, F., Srivastava, M., Preece, A., Julier, S., Rao, R. M., Kelley, T. D, Braines, D., Sensoy, M., Willis, C. J. & Gurram, P. (2017, August). Interpretability of deep learning models: A survey of results. In *2017 IEEE*

smartworld, ubiquitous intelligence & computing, advanced & trusted computed, scalable computing & communications, cloud & big data computing, Internet of people and smart city innovation (smartworld/SCALCOM/UIC/ATC/CBDcom/IOP/SCI) (pp. 1-6). IEEE.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., & Baroni, M. (2018). What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). 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), page 4171–4186, Minneapolis, Minnesota.

Ethayarajh, K. (2019). How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. *arXiv preprint arXiv:1909.00512.*

Ettinger, A., Elgohary, A., & Resnik, P. (2016, August). Probing for semantic evidence of composition by means of simple classification tasks. *In Proceedings of the 1st workshop on evaluating vector-space representations for nlp* (pp. 134-139).

Ettinger, A., Elgohary, A., Phillips, C., & Resnik, P. (2018). Assessing composition in sentence vector representations. *arXiv preprint arXiv:1809.03992.*

Farkas, R., Simon, E., Szarvas, G., & Varga, D. (2007). GYDER: maxent metonymy resolution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)* (pp. 161-164).

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis.*

Gritta, M., Pilehvar, M. T., Limsopatham, N., & Collier, N. (2017). Vancouver welcomes you! minimalist location metonymy resolution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1248-1259).

Gupta, A., Boleda, G., Baroni, M., & Padó, S. (2015, September). Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 12-21).

Harris, Z. S. (1954). Distributional structure. *Word* 10:146–162.

Hewitt, J., & Liang, P. (2019). Designing and Interpreting Probes with Control Tasks. *Proceedings of the 2019 Con.*

Johnson, M. (2009). How the Statistical Revolution Changes (Computational) Linguistics. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, pages 3-11.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. *Advances in neural information processing systems*, 28.

Kim, B., Khanna, R., & Koyejo, O. O. (2016). Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems, 29.*

Lakoff, G. & Johnson, M. (1980). *Metaphors We Live By*. United States: University of Chicago Press.

Landauer, T.K. & Dumais, S. (1997). A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychol. Rev.* 104:211–40.

Lenci, A., Montemagni, S., & Pirrelli, V. (2005). *Testo e computer. Introduzione alla linguistica computazionale*. Carocci editore.

Lenci, A. (2018). Distributional models of word meaning. *Annual review of Linguistics*, *4*, 151-171.

Li, H., Vasardani, M., Tomko, M., & Baldwin, T. (2020). Target Word Masking for Location Metonymy Resolution. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3696–3707, Barcelona, Spain (Online).

Littlemore, J. (2015). *Metonymy: Hidden Shortcuts in Language, Thought and Communication*. United Kingdom: Cambridge University Press.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Li, B., Zhou, H., He, J., Wang, M., Yang, Y., & Li, L. (2020). On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864.*

Markert, K. & Nissim, M. (2007). Semeval-2007 task 08: Metonymy resolution at semeval-2007. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 36–41.

Mathews, K. A. & Strube, M. (2020). A Large Harvested Corpus of Location Metonymy. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5678–5687, Marseille, France.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence, 267*, 1-38.

Mitchell, J. & Lapata, M. (2010). Composition in distributional models of semantics. *Cogn. Sci*. 34:1388–429

Nastase, V., Judea, A., Markert, K., & Strube, M. (2012). Local and global context for supervised and unsupervised metonymy resolution. *In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (pp. 183-193).

Nissim, M. & Markert, K. (2003). Syntactic features and word similarity for supervised metonymy resolution. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 56–63.

Pedinotti, P. & Lenci, A. (2020). Don't Invite BERT to Drink a Bottle: Modeling the Interpretation of Metonymies Using BERT and Distributional Representations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6831–6837, Barcelona, Spain (Online).

Pennington, J., Socher, R. & Manning, C. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532-1543.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. & Zettlemoyer, L. (2018). Deep Contextualized Word Representations, In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Poibeau, T. (2007). Up13: Knowledge-poor methods (sometimes) perform poorly. *In Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 418–421.

Pustejovsky, J. & Batiukova, O. (2019). *The Lexicon* (Cambridge Textbooks in Linguistics). Cambridge: Cambridge University Press.

Radden, G. & Kövecses, Z. (1999). Towards a theory of metonymy. In K.-U. Panther and G. Radden (eds.) *Metonymy in Language and Thought*. Amsterdam: John Benjamins, 17–59.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving language understanding by generative pre-training*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084.*

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144).

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, *8*, 842-866.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613-620.

Shwartz, V. & Dagan, I. (2019). Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7, 403-419.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vulić, I., Ponti, E. M., Litschko, R., Glavaš, G., & Korhonen, A. (2020). Probing pretrained language models for lexical semantics. *arXiv preprint arXiv:2010.05731*.

Zhang, W. & Gelernter, J. (2015). Exploring metaphorical senses and word representations for identifying metonyms. *arXiv preprint arXiv:1508.04515* .

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision* (pp. 19-27).

Appendix A

For each word pair in the dataset, a metonymic and a literal sentence are shown.

Word pair	**Metonymic example**	**Literal example**
fare bus	*This ticket will include your bus.*	*So Margot got off the bus.*
appointment doctor	*Close to 1 in 5 Americans have either considered skipping or actually skipped the doctor.*	*She went immediately to the doctor who had signed the death certificate.*
birth baby	*The affair with Karen continued after the baby.*	*She told me she was pregnant and asked if I would baptise the baby.*
boat sail	*Their 36 foot sail hit rough waters, losing power and the ability to steer.*	*Everyone goes out on longer boards with the biggest sails they can handle.*
car door	*The suspect drove away in a silver four door.*	*As I opened the door, a familiar tall figure, swinging a stick, strode past.*

car wheel	*Perhaps I just need to get used to driving a rear wheel.*	*Now there were more footsteps and the crunch of car wheels on gravel.*
door barn	*She heaved the barn open and disappeared in the gloom inside.*	*An endless walk it seemed to Gabriel, watching through the slatted door of the barn.*
guitar string	*It opens with his wistful, melancholy picking on a 12 string.*	*A good stringer will know the right string and tension for you.*
wool sheep	*They spun and wove their own garments from their undyed sheep.*	*Very stupid animals, sheep.*
ink pen	*He also walked around in public with pen smudged on his face.*	*Take the pen out of your nose dear, thank you!*
milk coconut	*Adults drink coconut, mixed with vodka, as an aperitif.*	*They'd have you carrying the coconuts on your head.*

book author	*This author that I purchased is really good so far.*	*One of the authors of that famous book was James Chadwick.*
author book	*The book "Purple Cow" argues that brands need to stand out.*	*Needless to say, whenever he could, he read the books in the original French.*
people ambulance	*He says he didn't see officers but saw the ambulance giving first aid.*	*Mr Chittenden was rushed to the Countess of Chester Hospital by ambulance.*
people boat	*This buoy was not seen by the boats as fog had now descended.*	*How is he coming, by train or boat?*
people building	*That building was being told to evacuate.*	*She walked around the building, trying to find the entrance.*
battery car	*I was so glad I had mine when my car went flat the other day.*	*I could visit my sister in Weymouth and buy a new car with the saved money.*

car battery	*These electric batteries can travel a couple of hundred kilometres on a single charge.*	*A battery is connected to the anode and cathode via leads A and C respectively.*
people motorcycle	*Dozens of motorcycles rode for their friends who were murdered six weeks ago.*	*A motorcycle was stolen from Turk Street, Alton on Wednesday night last week.*
pilot airplane	*Airplanes have long used checklists before take off to ensure safet.*	*Andalusia is very easy to reach by airplane.*
pilot helicopter	*Helicopters flying over the glacier also reported cracks in the glacier.*	*He can fly his helicopter at 100 kph.*
shoe toe	*If you can make it, wear work clothes and closed toes.*	*Tentatively, some dipped their toes to test the water.*
diaper baby	*It can be used as blanket or floor cover when changing the baby.*	*She told me she was pregnant and asked if I would baptise the baby.*

officer uniform	*Dark uniforms pursued him down the narrow alleys.*	*A man in a blue uniform was walking over towards them.*
woman beauty	*I know breathtaking beauties who pay their boyfriend's bills.*	*Then we fail to see the beauty in a leaf or in the clouds.*
chord guitar	*Bickert learned basic guitar from his older brother.*	*"I suppose he does know how to handle a guitar," said Miguel casually.*
music guitar	*He is a teacher, musician and composer of guitar.*	*"I suppose he does know how to handle a guitar," said Miguel casually.*
music harp	*His words will be accompanied by the traditional Celtic harp.*	*She could versify, play the harp, ride horseback, and sing.*
music piano	*As a composer, Haberbier was best known for his piano.*	*Business boomed, and Clementi sold countless new pianos.*

music violin	*The film uses intense violin in both its opening and closing credits.*	*Half the space is taken up by a picture of a violin -- ah yes, but one with a broken string.*
smoke cigar	*He exhales a puff of cigar and flashes what passes for a smile.*	*Then he stood and gathered his hat, cigar and white gloves together in one hand.*
smoke cigarette	*Nicotine affects the brain within seconds of inhaling a cigarette.*	*He knew the change would be slow, so he took out and lit a cigarette while he waited.*
sound keyboard	*Do you know how I felt when I heard your keyboard?*	*Thurston Moore of Sonic Youth also plays keyboard on one track.*
beer bottle	*Choristers swigged bottles and danced.*	*While you 're going upstairs get us a bottle of beer please?*
bottle beer	*Broken beers and cheap wine boxes are scattered around them.*	*Sometimes they want wine or beer with their lunches.*

carton milk	*In the morning, Dawn continues annoying Buffy by emptying the milk.*	*Ignoring him, she looked down to put milk and sugar into her cup.*
coffee mug	*I drink a couple of mugs in the morning and that's it for the day.*	*She took Zoe's mug and poured her fresh coffee.*
game ball	*During the day I will be playing ball or swimming with Gorm.*	*Whenever he gets the ball, everybody thinks something is about to happen.*
milk carton	*Play it safe and avoid food poisoning by discarding that expired carton.*	*It is also possible to use a large ice-cream carton, cut as shown in the drawing.*
mug coffee	*My coffee was still warm but half empty.*	*Rachel gave a weak smile and lifted up her cup for more coffee.*
bomb Hiroshima	*Sadako survived Hiroshima when she was only two years old.*	*It's like the little Japanese girl they found in the ruins of Hiroshima.*

vacation cottage	*Susan was feeling well and the family planned a cottage on Sparrow Lake for a week.*	*The cottage, he told Marshall, had been built the same time as the farm.*