Comparing Probabilistic, Distributional and Transformer-Based Models on Logical Metonymy Interpretation

Giulia Rambelli
University of Pisa
giulia.rambelli@phd.unipi.it

Emmanuele Chersoni
Hong Kong Polytechnic University
emmanuelechersoni@gmail.com

Alessandro Lenci
University of Pisa
alessandro.lenci@unipi.it

Philippe Blache
Aix-Marseille University
blache@lpl-aix.fr

Chu-Ren Huang
Hong Kong Polytechnic University
churen.huang@polyu.edu.hk

Abstract

In linguistics and cognitive science, *Logical metonymies* are defined as type clashes between an event-selecting verb and an entity-denoting noun (e.g. *The editor finished the article*), which are typically interpreted by inferring a hidden event (*e.g. reading*) on the basis of contextual cues.

This paper tackles the problem of logical metonymy *interpretation*, that is, the retrieval of the covert event via computational methods. We compare different types of models, including the probabilistic and the distributional ones previously introduced in the literature on the topic. For the first time, we also tested on this task some of the recent Transformer-based models, such as BERT, RoBERTa, XLNet, and GPT-2.

Our results show a complex scenario, in which the best Transformer-based models and some traditional distributional models perform very similarly. However, the low performance on some of the testing datasets suggests that logical metonymy is still a challenging phenomenon for computational modeling.

1 Introduction

The phenomenon of *logical metonymy* is defined as a type clash between an event-selecting metonymic verb (e.g., *begin*) and an entity-denoting nominal object (e.g., *the book*), which triggers the recovery of a hidden event (e.g., *reading*). Logical metonymies have been widely studied, on the one hand, in theoretical linguistics as they represent a challenge to traditional theories of compositionality (Asher, 2015; Pustejovsky and Batiukova, 2019). On the other hand, they received extensive attention in cognitive research on human sentence processing as they determine extra processing costs during online sentence comprehension (McElree et al., 2001; Traxler et al., 2002), apparently related to "the deployment of operations to construct a semantic representation of the event" (Frisson and McElree, 2008).[1]

Logical metonymy has also been explained in terms of the *words-as-cues hypothesis* proposed by Jeffrey Elman (Elman, 2009, 2014). This hypothesis relies on the experimental evidence that human semantic memory stores knowledge about events and their typical participants (see McRae and Matsuki (2009) for an overview) and claims that words act like cues to access event knowledge, incrementally modulating sentence comprehension. The results obtained in a probe recognition experiment by Zarcone et al. (2014), in line with this explanation, suggest that speakers interpret logical metonymies by inferring the most likely event the sentences could refer to, given the contextual cues. Previous research in NLP on logical metonymy has often been influenced by such theoretical explanation (Zarcone and Padó, 2011; Zarcone et al., 2012; Chersoni et al., 2017).

In our contribution, we propose a general comparison of different classes of computational models for logical metonymy. To begin with, we tested two approaches that have been previously introduced in the literature on the topic: probabilistic and distributional models (Zarcone et al., 2012). We also examined the Structured Distributional Model (SDM) by Chersoni et al. (2019), which represents sentence meaning with a combination of formal structures and distributional embeddings to dynamically integrate knowledge about events and their typical participants, as they are activated by lexical items. Finally, to the best of our knowledge, we are the first ones to include the recent Transformer language models into a contrastive study on

[1] Notice however that the evidence is not uncontroversial: Delogu et al. (2017) report that coercion costs largely reflect word surprisal, without any specific effect of type shift in the early processing measures.

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 224–234

logical metonymy. Transformers (Vaswani et al., 2017; Devlin et al., 2019) are the dominant class of NLP systems in the last few years, since they are able to generate "dynamic" representations for a target word depending on the sentence context. As the interpretation of logical metonymy is highly sensitive to context, we deem that the contextual representations built by Transformers might be able to integrate the covert event that is missing in the surface form of the sentence.

All models are evaluated on their capability of **assigning the correct interpretation to a metonymic sentence**, that is, **recovering the verb that refers to the correct interpretation**. This task is hard for computational models, as they must exploit contextual cues to distinguish covert events with a high typicality (e.g., *The pianist begins the symphony* → *playing*) from plausible but less typical ones (→ *composing*).

2 Related Work

2.1 Computational Models of Logical Metonymy

According to Zarcone et al. (2013), the phenomenon of logical metonymy can be explained in terms of the *thematic fit*, that is, the degree of compatibility between the verb and one of its arguments (the direct object, in this case). On the one hand, a low thematic fit between an event-selecting verb and an entity-denoting argument triggers the recovery of a covert event, while on the other hand, the recovered event is often the best fitting one, given the information available in the sentence.

Research in NLP on logical metonymy initially focused on the problem of covert event retrieval, which was tackled by means of probabilistic models (Lapata and Lascarides, 2003; Shutova, 2009), or by using Distributional Semantic Models (DSMs) that identify the candidate covert event with the one that has the highest thematic fit with the arguments in the sentence (Zarcone et al., 2012). Following the psycholinguistic works by McElree et al. (2001) and Traxler et al. (2002), which reported increased reading times and longer fixations in eye-tracking for the metonymic sentences, Zarcone et al. (2013) proposed a distributional model of the thematic fit between verb and object, and showed that it accurately reproduces the differences between the experimental conditions in the data from the two original studies.

A general distributional model for sentence comprehension was used by Chersoni et al. (2017) to simultaneously tackle both these two aspects of logical metonymy (covert event retrieval and increased processing times), although at the cost of a highly-elaborated compositional model. The authors recently introduced a more up-to-date and refined version of their sentence comprehension model (Chersoni et al., 2019), but it has not been tested on the logical metonymy task so far.

2.2 Transformer Models in NLP

The traditional approach in Distributional Semantics has been the building of a single, stable vector representation for each word type in the corpus (Turney and Pantel, 2010; Lenci, 2018). Lately, a new generation of embeddings has emerged, in which each occurrence of a word in a specific sentence context gets a unique representation (Peters et al., 2018). The most recent systems typically rely on an LSTM or a Transformer architecture for getting word representations: they are trained on large amounts of textual data and the word vectors are learned as a function of the internal states of the encoder, such that a word in different sentence contexts determines different activation states and is represented by a different vector. Thus, embeddings generated by these new models are said to be *contextualized*, as opposed to the *static* vectors generated by the earlier frameworks, and they aim at modeling the specific sense assumed by the word in context. One of the most popular and successful contextualized model is probably BERT (Devlin et al., 2019), whose key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modelling. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training. The results of the paper show that a language model with bidirectional training can have a deeper sense of language context and structure than single-direction language models.

An interesting aspect of Transformer models like BERT is that they are trained via **masked language modeling**, that is, they have to retrieve a word that has been masked in a given input sentence. Since interpreting logical metonymy implies the retrieval of an event that is not overtly expressed and that humans retrieve integrating the lexical cues in the sentence, these models are potentially a very good fit for this task. To draw an analogy, we could

imagine that the covert event is a verb that has been 'masked' in the linguistic input and that we ask BERT-like models to make a guess.

It is important to point out that not all Transformers are used for masked language modeling: among those tested for this study, BERT and RoBERTa are directly trained with this objective, XLNet is trained with permutation language modeling, but can still retrieve a hidden word given a bidirectional context, and GPT-2 works similarly to a traditional, unidirectional language model.

3 Experimental Settings

3.1 Task

Our research question focuses on how computational models can interpret metonymic sentences. To explore this issue, we define the task of logical metonymy interpretation as a **covert event recovery** task. More specifically, given a sentence like *The architect finished the house*, the computational model has to return the most likely hidden verb for the sentence, i.e. the covert event representing its interpretation. Despite the architectural differences, all tested models compute a *plausibility score* of a verb as expressing the covert event associated with a <subject, metonymic verb, object> triple. We evaluate the scores returned by a model against human judgments using the standard measures of accuracy and correlation depending if the dataset contains categorical or continuous variables.

3.2 Datasets

In our experiments, we use three datasets designed for previous psycholinguistic studies, and a newly created one by means of an elicitation task.

The **McElree** dataset (**MC**) comprises the stimuli from the sentences of the self-paced reading experiment of McElree et al. (2001) and includes 30 pairs of tuples. Each pair has the same subject, metonymic verb, object, just the covert verb varies. As in the conditions of the original experiment, the hidden verb could be either highly plausible, or plausible but less typical, given the subject and the object of the tuple. The **Traxler** dataset (**TR**) results from the sentences of the eye-tracking experiment of Traxler et al. (2002) and includes 36 pairs of tuples. The format is the same as the McElree dataset. On these two datasets, the models have to perform a *binary classification task*, with the goal of assigning a higher score to the covert event in the typical condition.

The **Lapata-Lascarides** dataset (**L&L**) (Lapata and Lascarides, 2003) includes 174 tuples, each composed by a metonymic verb, an object and a potential covert verb. The authors collected plausibility ratings for each metonymy by turning the tuples into sentences and used the Magnitude Estimation Paradigm (Stevens, 1957) to ask human subjects to rate the plausibility of the interpretation of the metonymic verb. Finally, the mean ratings have been normalized and log-transformed.

A further dataset of **recovered covert events** (**CE**) was collected by the authors. The metonymic sentences used in the McElree and Traxler experiments were turned into 69 templates with an empty slot corresponding to the covert event (e.g., *The student began ___ the book late in the semester*). Thirty subjects recruited with crowdsourcing were asked to produce two verbs that provided the most likely fillers for the event slot. Out of the $4,084$ collected verbs, we selected those with a production frequency ≥ 3 for a given stimulus. The final dataset comprises 285 items each consisting of a subject – metonymic verb – object tuple t and a covert event e associated with a salience score corresponding to the event conditional probability given the tuple $P(e|t)$ (i.e., the production frequency of e normalized by the total events produced for t). In the case of the latter two datasets, for each model we compute the Spearman's correlation between the probabilities generated by the model and the human judgements. Examples from these datasets are provided in Table 1.

While collecting the data for **CE**, we also run a statistical comparison between the production frequencies of the verbs in the typical and in the atypical condition that appear in the binary classification datasets, to ensure that humans genuinely agree on the higher typicality of the former. The result confirmed this assumption: according to the Wilcoxon signed rank test with continuity correction, the frequencies of production of the typical verbs for the **MC** dataset were significantly higher ($W = 424, p < 0.001$), and the same holds for the typical verbs in the **TR** dataset ($W = 526.5, p < 0.001$).

3.3 Models

In the following section, we describe the general aspects of the computational models that we tested on logical metonymy interpretation.

Dataset	Subject-verb-object	Covert event	Condition/Score	Size
MC	chef start dinner	*prepare* *eat*	HIGH_TYP LOW_TYP	30 (pairs)
TR	dieter resist cake	*eat* *taste*	HIGH_TYP LOW_TYP	36 (pairs)
L&L	— start experiment	*implement* *study*	0.1744 0.0184	174
CE	architect start house	*draw* *build*	0.348 0.087	258

Table 1: Examples of stimuli from each dataset.

3.3.1 Probabilistic Model

As a baseline model, we adopt the simple probabilistic approach proposed by Lapata and Lascarides (2003) and replicated by Zarcone et al. (2012) as the $\mathbf{SO_p}$ model, which was reported as the best performing probabilistic model on the task. The interpretation of a logical metonymy (e.g., *The pianist began the symphony*) is modelled as the joint distribution *P(s, v, o, e)* of the variables *s* (the subject, *pianist*), *v* (the metonymic verb, *began*), *o* (the object, *symphony*), and the covert event *e* (e.g., *play*). We compute that probability considering the metonymic verb constant:

$$P(s, v, o, e) \approx P(e)P(o|e)P(s|e)$$

The verb E representing the preferred interpretation of the metonymy is the verb e maximizing the following equation:

$$E = argmax_e P(e)P(o|e)P(s|e)$$

We computed the statistics from a 2018 dump of the English Wikipedia, parsed with the Stanford CoreNLP toolkit (Manning et al., 2014).

Dataset	Coverage
MC	19/30 (pairs)
TR	21/36 (pairs)
L&L	151/174 (items)
CE	195/285 (items)

Table 2: Coverage for the probabilistic model.

3.3.2 Logical Metonymy as Thematic Fit

Distributional models of logical metonymy assume that the event recovery task can be seen as a *thematic fit* task: recovering the covert event means identifying the verb with the highest thematic fit with the metonymic sentence. We reimplement the distributional model by Zarcone et al. (2012) with the following procedure:

- we retrieve the n $(= 500)$[2] most strongly associated verbs for the subject and the object respectively, and we take the intersection of the two lists;
- we update their association scores using either the sum (add) or the product $(prod)$ function;
- we select the embeddings corresponding to the first m $(= 20)$ verbs in this list and we add them together to create the prototype vector of the verb given the subject and the object;
- the thematic fit of the covert event *e* with respect to the nominal entities is computed as the similarity score of its corresponding lexical vector $\vec{e}$ with the prototype vector. As we did the probabilistic model, we discard the metonymic verb from this computation. [3]

We test two variations of this model, **TF-add** and **TF-prod**, which differ for the filler selection update function. Statistics were extracted from Wikipedia 2018, and the vectors were the publicly-available Wikipedia embeddings [4] trained with the FastText model (Bojanowski et al., 2017). The verb-filler association score is the Local Mutual Information (Evert, 2008). Similarly, the scores for the subject fillers are defined as:

$$LMI(s, e) = f(e \xleftarrow{sbj} s) log_2 \frac{p(s|e)}{p(s)p(e)}$$

[2] We set a high value for this parameter in order to maximize the coverage.

[3] Zarcone et al. (2012) show that, for both the probabilistic and the distributional model, including the metonymic verb does not help too much in terms of performance and leads to coverage issues.

[4] https://fasttext.cc/docs/en/english-vectors.html

where s is the subject, e the covert event, and $f(e \xleftarrow{sbj} s)$ indicates the frequency of e with the subject. The scores for the object position are computed with the following formula:

$$LMI(o,e) = f(e \xleftarrow{obj} o) log_2 \frac{p(o|e)}{p(o)p(e)}$$

where o is the object and $f(e \xleftarrow{obj} o)$ represents the joint frequency of e with the object.

3.3.3 Structured Distributional Model

The **Structured Distributional Model** (SDM) proposed by Chersoni et al. (2019) consists of two components: a *Distributional Event Graph* (henceforth, DEG), and a meaning composition function. DEG represents event knowledge as a graph automatically built from parsed corpora, where the nodes are words associated to a numeric vector, and the edges are labeled with syntactic relations and weighted using statistic association measures. Each event is represented as a path in DEG, that is, a sequence of edges (relations) which joins a sequence of vertices (words). Thus, given a lexical cue w, it is possible to identify the associated events and to generate expectations about incoming inputs on both the paradigmatic and the syntagmatic axis.

The composition function makes use of two semantic structures (inspired by DRT (Kamp, 2013)): the *linguistic condition* (LC), a context-independent tier of meaning, and the *active context* (AC), which accumulates contextual information available during sentence processing or activated by lexical items. The crucial aspect is that the model associates a vectorial representation to these formal structures: $\vec{LC}$ is the sum of the embeddings of the lexical items of a sentence; $\vec{AC}$, for each syntactic slot, is represented as the centroid vector built out of the role vectors $\vec{r_1}, ..., \vec{r_n}$ available in AC, i.e. the syntactic associates of the lexical items that have been already processed.

In our implementation of SDM, the DEG is constructed by extracting syntactic relations from the same dump of Wikipedia adopted in the previous models, and we chose as lexical embeddings the same FastText Wikipedia vectors. Following the same assumption of the previous experiment, we model the covert event recovery task as a thematic fit task: the goal is to predict the hidden verb on the basis of the subject and the object, treating the metonymic verb as a constant. Specifically, the model builds a semantic representation for each tuple in the dataset. The linguistic condition vector $\vec{LC}$ contains the sum of the subject and object embeddings. At the same time, the event knowledge vector $\vec{AC}$ contains the prototypical embedding for the main verb, using DEG to retrieve the most associated verbs for the subject and the object, as in Chersoni et al. (2019). The scoring function has been adapted to the event recovery task as follows:

$$cos(\vec{e}, \vec{LC}(sent)) + cos(\vec{e}, \vec{AC}(sent))$$

where *sent* refers to the metonymic test tuple. In other words, we quantify the typicality of a verb for a tuple subject-object as the sum of i.) the cosine similarity between the event embedding and the additive combination of the other argument vectors ($\vec{LC}$) and ii.) the cosine similarity between the event embedding and the prototype vector representing the active context ($\vec{AC}$).

	Model settings				Data size
	L	H	A	P	
BERT large-cased	24	1024	16	340M	16GB
RoBERTa large	24	1024	16	355M	160GB
XLNet large-cased	24	1024	16	340M	113GB
GPT-2 extra-large	48	1600	25	1542M	40 GB

Table 3: Comparison between transformer models. Model details: *L*: number of layers, *H*: dimension of hidden states, *A*: attention head numbers, and *P*: total parameter size.

3.3.4 Transformer-based Models

We experiment with four Transformer models which have been shown to obtain state-of-the-art performances on several NLP benchmarks.

The popular **BERT** model (Devlin et al., 2019) was the first to adopt the bidirectional training of Transformer for a language modeling task. To make this kind of training possible, BERT introduced a masked language modeling objective function: random words in the input sentences are replaced by a [MASK] token and the model attempts to predict the masked token based on the surrounding context. Simultaneously, BERT is optimized on a next sentence prediction task, as the model receives sentence pairs in input and has to predict whether the second sentence is subsequent to the

first one in the training data.[5] BERT has been trained on a concatenation of the BookCorpus and the English Wikipedia, for a total of 3300M tokens ca. In our experiments, we used the larger pre-trained version, called BERT-large-cased.

RoBERTa (Liu et al., 2019) has the same architecture as BERT, but it introduces several parameter optimization choices: it makes use of dynamic masking (compared to the static masking of the original model), of a larger batch-size and a larger vocabulary size. Moreover, the input consists of complete sentences randomly extracted from one or multiple documents, and the next sentence prediction objective is removed. Besides the optimized design choice, another key difference of RoBERTa with the other models is the larger training corpus, which consists of a concatenation of the BookCorpus, CCNEWS, OpenWebText, and STORIES. With a total 160GB of text, RoBERTa has access to more potential knowledge than the other models. For our tests, we used the large pre-trained model.

XLNet (Yang et al., 2019) is a generalized autoregressive (AR) pretraining method which uses the context words to predict the next word. The AR architecture is constrained to a single direction (either forward or backwards), that is, context representation takes in consideration only the tokens to the left or to the right of the i-th position, while BERT representation has access to the contextual information on both sides. To capture bidirectional contexts, XLNet is trained with a permutation method as language modeling objective, where all tokens are predicted but in random order. XLNet's training corpora were the same as BERT plus Giga5, ClueWeb 2012-B and Common Crawl, for a total of 32.89B subword piece. Also in this case, we used the large pre-trained model.

GPT-2 (Radford et al., 2019), a variation of GPT, is a uni-directional transformer language model, which means that the training objective is to predict the next word, given all of the previous words. Compared with GPT, GPT-2 optimizes the layer normalization, expands the vocabulary size to 50,257, increases the context size from 512 to 1024 tokens, and optimizes with a larger batch size of 512. In addition, GPT-2 is pre-trained on WebText, which was created by scraping web pages, for a total of 8 million documents of data (40 GB). We used the XL version of GPT-2 for our experiments.

The parameters of the Transformer models are reported in Table 3. BERT, RoBERTa and XLNet are used to perform a *word prediction task*: given a sentence and a masked word in position k, they compute the probability of a word w_k given the $context_k$: $P(w_i|context_k)$. For our experiments, the context is the entire sentence S with the k-th word (the covert event) being replaced by a special token '[MASK]'. Therefore, we turned the test tuples into full sentences, masking the verb as in the example below: *The architect finishes [MASK] house.* [6] We then compute the probability of a hidden verb to occur in that position, and we expect the preferred verb to get a high value. We performed this task using the packages of the HappyTransformer library.[7]

As GPT-2 works as a traditional language model, we adopted this model to calculate the probability of the entire sentence (instead of the probability of the hidden verb given the context). In this case, we expect that sentences evoking more typical events get higher values. We adopted the lm-scorer package to compute sentence probabilities.[8]

4 Evaluation Results

Table 5 and 4 report the final evaluation scores. The performance of the probabilistic model is in line with previous studies, and it outperforms distributional models in some cases, proving that it is indeed a hard baseline to beat. However, accuracy and correlation are computed only on a subgroup of the test items: actually, the model covers about 60% of the datasets' tuples (86.8% for L&L), as we reported in Table 2. Coverage is the main issue probabilistic models have to face (Zarcone et al., 2013), while distributional models do not experience such limitation.

Regarding the thematic fit models, we observe that there is no difference between the TF-add and TF-prod models, as they obtain similar scores.

[5] Notice that the usefulness of this secondary objective function was questioned, and it was indeed removed in more recent models (Yang et al., 2019; Liu et al., 2019; Joshi et al., 2020).

[6] One of the anonymous reviewers argues that the performance of the Transformer-based models might be influenced by the prompt sentence and suggest more variations of the input sentences. We indeed tested several manipulations of the inputs before feeding them to the transformers, changing 1) the tense of the metonymic verb (using the past tense) and 2) the number of the direct object (we used the plurals of the dataset nouns). However, the results did not show any consistent trend.

[7] https://github.com/EricFillion/happy-transformer

[8] https://pypi.org/project/lm-scorer/

	Probabilistic	Distributional			Transformer-based			
	SOp	*TF-add*	*TF-prod*	*SDM*	*BERT*	*RoBERTa*	*XLNet*	*GPT-2*
MC	0.68	0.70	0.73	0.77	0.70	0.80	0.40	**0.87**
TR	0.48	0.53	0.53	**0.72**	0.47	**0.72**	0.39	0.69
O. P.	0.58	0.62	0.63	0.75	0.59	0.76	0.40	**0.78**

Table 4: Results for binary classification task.

	Probabilistic	Distributional			Transformer-based			
	SOp	*TF-add*	*TF-prod*	*SDM*	*BERT*	*RoBERTa*	*XLNet*	*GPT-2*
L&L	0.53	0.41	0.41	0.53	0.61	**0.73**	0.04	0.43
CE	0.36	0.26	0.22	**0.40**	0.27	0.39	0.18	0.31
O. P.	0.45	0.34	0.32	0.47	0.44	**0.56**	0.11	0.37

Table 5: Results for correlation task.

However, we need to point out that, when the system computes the intersection of the two lists of the top verbs for subjects and objects, sometimes the number of retrieved items is less than 20 (the model parameter for the verb embedding selection, cf. Section 3.3.2). Therefore, independently of the selected function, the verbs used to compute the prototypical vector are eventually all those belonging to the intersection. Moreover, TF-models are often close to, and never significantly outperform the probabilistic baseline.

Among the distributional models, SDM is the one that obtains a considerable performance across all the datasets. This model performs close to RoBERTa both in the Traxler and in the CE dataset. This result is surprising, considering that SDM is trained just on a dump of Wikipedia, while RoBERTa is trained on 160 GB of text and implements advanced deep learning techniques. This outcome confirms that SDM, which has been designed to represent event knowledge and the dynamic construction of sentence meaning, is able to adequately model the typicality of events. This aspect has been suggested to be one of the core components of the language processing system (Baggio and Hagoort, 2011; Baggio et al., 2012; Chersoni et al., 2019).

On the other hand, Transformers also provided interesting results. RoBERTa achieves the best score for the L&L dataset, reaching a statistical significance of the improvement over SDM ($p < 0.01$).[9] More importantly, it is the only Transformer that consistently obtains good results across all datasets, while the scores from other Transformer models are highly fluctuating. We believe that the gigantic size of the training corpus is a factor that positively affects its performance. At the same time, GPT-2 achieves the highest score for MC dataset (0.87) (but the improvement over RoBERTa and SDM does not reach statistical significance), although it performs significantly lower on the other benchmarks[10].

For the sake of completeness, we also report the overall performance of each model over the two tasks. Results identify RoBERTa and GPT-2 as the best models for the correlation and classification tasks, respectively. However, we wonder if the average score is a valid measure to identify the best model. These two models tend to have a wavering behavior, which results in large differences between the two datasets scores. Specifically, Roberta achieves 0.75 for the L&L dataset, but only 0.39 for the CE one, with 0.36 points of difference. Similarly, GPT-2 reaches 0.89 scores for the MC dataset, but its performance goes down by 0.16. On the contrary, SDM behavior is more stable, with a smaller gap between the two datasets' scores (0.13 point difference for the correlation task and just 0.05 for the accuracy task).

4.1 Error analysis

Binary classification task For the MC and TR datasets, we evaluate the models for their capability of assigning a higher probability to the verb in the typical condition. It is important to empha-

[9] The p-value is computed with Fisher's r-to-z transformation, one-tailed test.

[10] We determine the significance of differences between models for MC and TR datasets with a McNemar's Chi-Square Test, applied to a 2x2 contingency matrix containing the number of correct and incorrect answers (replicating the approach of Zarcone et al. (2012)).

size that both verbs are plausible in the context, but one describes a more likely event given the subject and the object. This remark is essential, because it explains the performance of all models, distributional and Transformer ones.

To identify which tuples are the most difficult ones, we built a heat map visualizing the correctly-predicted ones in blue, and the wrong ones in yellow (see Figures 1 and 2). We do not consider the accuracy values obtained by the probabilistic model for its partial coverage.

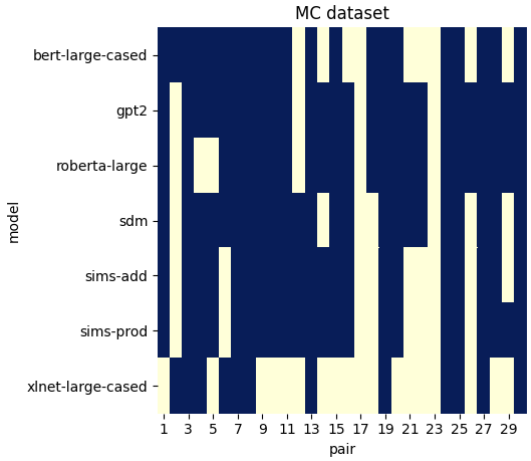

Figure 1: Heat map for error analysis over MC dataset.

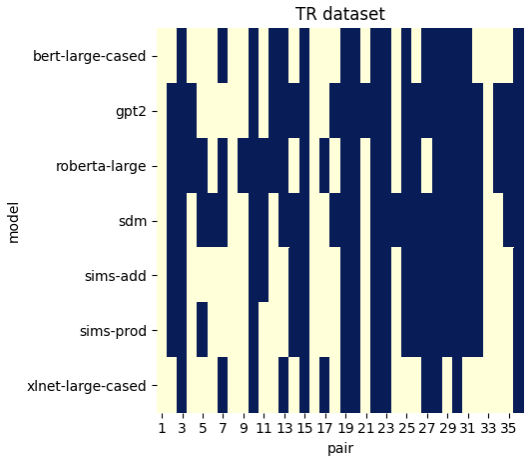

Figure 2: Heat map for error analysis over TR dataset.

This visualization technique reveals that some pairs are never predicted correctly, corresponding to the fully vertical yellow lines in the figures. In what follow we report the tuples that are consistently mistaken for MC (1) and TR (2) datasets.

(1) a. *The teenager starts the novel.*
b. *The worker begins the memo.*

(2) a. *The editor finishes the newspaper.*
b. *The director starts the script.*
c. *The teenager begins the novel.*

In all the above cases, a model must discriminate between the verb *read* (HIGH_TYP) and *write* (LOW_TYP).[11] It is interesting to notice that, for many of the *read-write* pairs in the binary classification data, the production frequencies of typical and atypical verb are much closer than on average, suggesting that the interpretation requires understanding of subtle nuances of context-sensitive typicality, which might not be trivial even for humans.

Furthermore, in Figure 2 we observe that for two TR's pairs, SDM is the only one picking the right choice: *The stylist starts the braid* and *The auditor begins the taxes*. It seems that models regularly tend to prefer a verb with a more generic and undetermined meaning (*make* and *do*, respectively), while only SDM correctly assigns the HIGH_TYP class to the verbs that indicate more precisely the manner of doing something (*braid* and *audit*).

On the other hand, GPT-2 and RoBERTa managed to pick the right choice for a few of the *read-write* items on which SDM is mistaken.

Correlation task Correlation is a more complex task compared to classification, as the lower scores also reveal. To better understand our results, we select the best model for the CE (i.e., SDM) and L&L (i.e., RoBERTa) datasets, and we plot the linear relationship between the human ratings and the model-derived probabilities.[12] For CE, Figure 3 reveals 1) a small positive correlation between the two variables, 2) a large amount of variance, and 3) a few outliers.

As for L&L in Figure 4, the majority of the points follow a roughly linear relationship, and there is a small variation around the trend. Nevertheless, this result could be influenced by the form of the input sentences. For all the other datasets, we masked the token between the verb and the object, and the corresponding hidden verb had to be in the progressive form (*The chef starts [cooking] dinner*). For L&L, instead, we chose to insert the preposition *to* after the verb since lots of the metonymic verbs (*want*, *try*, etc.) require to be followed by the infinitive verb. Thus, the context gives a higher

[11] Except for the sentence in 2.a, where the typical verb is *edit*.

[12] We apply the logarithmic transformation of data for visualization purposes.

probability to verbs as masked tokens, while different parts of speech could be equally plausible for the other conditions.

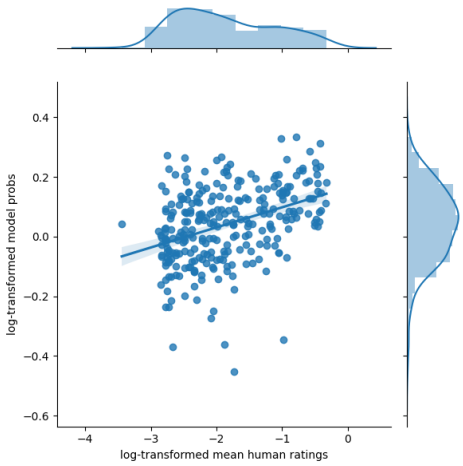

Figure 3: SDM correlation for CE.

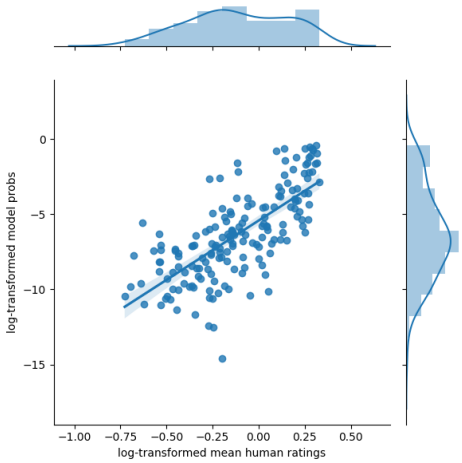

Figure 4: RoBERTa correlation for L&L.

5 Discussion and Conclusions

In this paper, we have presented a comparative evaluation of several computational models on the task of logical metonymy interpretation. We frame this problem as the retrieval of an event that is not overtly expressed in the surface form of the sentence. According to Elman's Words-as-Cues framework, human subjects can infer the covert event in logical metonymy thanks to the generalized knowledge about events and participants stored in their semantic memory. Hence, during sentence processing, words in the sentence create a network of mutual expectations that triggers the retrieval of typical events associated with lexical items and generates expectations about the upcoming words (Elman, 2014). To tackle the task of logical metonymy interpretation, computational models must be able to recover unexpressed relationships between the words, using a context-sensitive representation of meaning that captures this event knowledge.

The most compelling outcome of the reported experiments is probably the performance of SDM, which achieves the best score for the TR and the CE datasets. These results demonstrate the significance of encoding event structures *outside* the embeddings (which are treated as nodes in a distributional graph), and the ability of the SDM compositional function to dynamically update the semantic representation for a sentence. However, the evaluation scores are not very high, especially in the correlation task. Results reveal that the contextualized information used by computational models is useful to recall plausible events connected to the arguments, but this is still not sufficient. Even Transformer models, which currently report state-of-the-art performances on several NLP benchmarks, are not performing significantly better than the SDM model, which is trained on a smaller corpus and without any advanced deep learning technique. Error analysis highlights that they are able to identify the plausible scenarios in which the participants could occur, but they still struggle in perceiving different nuances of typicality. Our experiments show how the logical metonymy task can be seen as a testing ground to check whether computational models encode common-sense event knowledge.

Future work might follow two directions. On the one hand, expanding the coverage of the graph could favourably increase the performance of SDM. On the other hand, Transformer models could be tested with new experimental settings, such as the fine-tuning of the pre-trained weights on thematic fit-related (Lenci, 2011; Sayeed et al., 2016; Santus et al., 2017) or semantic role classification tasks (Collobert et al., 2011; Zapirain et al., 2013; Roth and Lapata, 2015).

6 Acknowledgements

This work, carried out within the Institut Convergence ILCB (ANR-16-CONV-0002), has benefited from support from the French government, managed by the French National Agency for Research (ANR) and the Excellence Initiative of Aix-Marseille University (A*MIDEX). We thank the anonymous reviewers for their insightful feedback.

References

Nicholas Asher. 2015. Types, Meanings and Coercions in Lexical Semantics. *Lingua*, 157:66–82.

Giosuè Baggio and Peter Hagoort. 2011. The Balance between Memory and Unification in Semantics: A Dynamic Account of the N400. *Language and Cognitive Processes*, 26(9):1338–1367.

Giosuè Baggio, Michiel Van Lambalgen, and Peter Hagoort. 2012. The Processing Consequences of Compositionality. *The Oxford Handbook of Compositionality. Oxford*, pages 657–674.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Emmanuele Chersoni, Alessandro Lenci, and Philippe Blache. 2017. Logical Metonymy in a Distributional Model of Sentence Comprehension. In *Proceedings of *SEM*.

Emmanuele Chersoni, Enrico Santus, Ludovica Pannitto, Alessandro Lenci, Philippe Blache, and C-R Huang. 2019. A Structured Distributional Model of Sentence Meaning and Processing. *Natural Language Engineering*, 25(4):483–502.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Francesca Delogu, Matthew W Crocker, and Heiner Drenhaus. 2017. Teasing Apart Coercion and Surprisal: Evidence from Eye-Movements and ERPs. *Cognition*, 161:46–59.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019*, Minneapolis, MN.

Jeffrey L Elman. 2009. On the Meaning of Words and Dinosaur Bones: Lexical Knowledge without a Lexicon. *Cognitive Science*, 33(4):547–582.

Jeffrey L Elman. 2014. Systematicity in the Lexicon: On Having your Cake and Eating It Too. In Paco Calvo and John Symons, editors, *The Architecture of Cognition: Rethinking Fodor and Pylyshyn's Systematicity Challenge*. The MIT Press, Cambridge, MA.

Stefan Evert. 2008. Corpora and collocations. *Corpus linguistics. An international handbook*, 2:1212–1248.

Steven Frisson and Brian McElree. 2008. Complement Coercion is not Modulated by Competition: Evidence from Eye Movements. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 34(1):1–11.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Hans Kamp. 2013. *Meaning and the Dynamics of Interpretation: Selected Papers by Hans Kamp*. Brill, Leiden-Boston.

Mirella Lapata and Alex Lascarides. 2003. A Probabilistic Account of Logical Metonymy. *Computational Linguistics*, 29(2):261–315.

Alessandro Lenci. 2011. Composing and Updating Verb Argument Expectations: A Distributional Semantic Model. In *Proceedings of the ACL Workshop on Cognitive Modeling and Computational Linguistics*.

Alessandro Lenci. 2018. Distributional Models of Word Meaning. *Annual Review of Linguistics*, 4:151–171.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Brian McElree, Matthew J Traxler, Martin J Pickering, Rachel E Seely, and Ray Jackendoff. 2001. Reading Time Evidence for Enriched Composition. *Cognition*, 78:B17–B25.

Ken McRae and Kazunaga Matsuki. 2009. People Use their Knowledge of Common Events to Understand Language, and Do So as Quickly as Possible. *Language and Linguistics Compass*, 3(6):1417–1429.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of NAACL*.

James Pustejovsky and Olga Batiukova. 2019. *The Lexicon*. Cambridge University Press.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models Are Unsupervised Multitask Learners. *OpenAI Blog*, 1(8):9.

Michael Roth and Mirella Lapata. 2015. Context-Aware Frame-Semantic Role Labeling. *Transactions of the Association for Computational Linguistics*, 3:449–460.

Enrico Santus, Emmanuele Chersoni, Alessandro Lenci, and Philippe Blache. 2017. Measuring Thematic Fit with Distributional Feature Overlap. In *Proceedings of EMNLP*.

Asad Sayeed, Clayton Greenberg, and Vera Demberg. 2016. Thematic Fit Evaluation: An Aspect of Selectional Preferences. In *Proceedings of the ACL Workshop on Evaluating Vector Space Representations for NLP*.

Ekaterina Shutova. 2009. Sense-Based Interpretation of Logical Metonymy Using a Statistical Method. In *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, pages 1–9.

Stanley S Stevens. 1957. On the Psychophysical Law. *Psychological review*, 64(3):153.

Matthew J Traxler, Martin J Pickering, and Brian McElree. 2002. Coercion in Sentence Processing: Evidence from Eye-Movements and Self-Paced Reading. *Journal of Memory and Language*, 47(4):530–547.

Peter D Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237*.

Benat Zapirain, Eneko Agirre, Lluis Marquez, and Mihai Surdeanu. 2013. Selectional Preferences for Semantic Role Classification. *Computational Linguistics*, 39(3):631–663.

Alessandra Zarcone, Alessandro Lenci, Sebastian Padó, and Jason Utt. 2013. Fitting, not Clashing! A Distributional Semantic Model of Logical Metonymy. In *Proceedings of IWCS*.

Alessandra Zarcone and Sebastian Padó. 2011. Generalized Event Knowledge in Logical Metonymy Resolution. In *Proceedings of CogSci*.

Alessandra Zarcone, Sebastian Padó, and Alessandro Lenci. 2014. Logical Metonymy Resolution in a Words-as-Cues Framework: Evidence from Self-paced Reading and Probe Recognition. *Cognitive Science*, 38(5):973–996.

Alessandra Zarcone, Jason Utt, and Sebastian Padó. 2012. Modeling Covert Event Retrieval in Logical Metonymy: Probabilistic and Distributional Accounts. In *Proceedings of the NAACL Workshop on Cognitive Modeling and Computational Linguistics*.