Metaphor Interpretation Using Word Embeddings

Kfir Bar, Nachum Dershowitz, Lena Dankin

School of Computer Science, Tel Aviv University

Abstract. We suggest a model for metaphor interpretation using word embeddings trained over a relatively large corpus. Our system handles nominal metaphors, like *time is money*. It generates a ranked list of potential interpretations of given metaphors. Candidate meanings are drawn from collocations of the topic (*time*) and vehicle (*money*) components, automatically extracted from a dependency-parsed corpus. We explore adding candidates derived from word association norms (common human responses to cues). Our ranking procedure considers similarity between candidate interpretations and metaphor components, measured in a semantic vector space. Lastly, a clustering algorithm removes semantically related duplicates, thereby allowing other candidate interpretations to attain higher rank. We evaluate using different sets of annotated metaphors, with encouraging preliminary results.

1 Introduction

Metaphor is pervasive in language and thought [3]. Based on a quantitative analysis, Krennmayr [6] found that even in academic papers almost every fifth word is part of a metaphorical concept, broadly construed.

Already Aristotle analyzed and wrote about the use of metaphor. "Metaphor", he says in the *Poetics*, "consists in giving the thing a name that belongs to something else." *The sunset of life* is one of his examples. In *Rhetoric* he explains: "A simile is also a metaphor; for there is little difference: when the poet says, 'He rushed as a lion,' it is a simile, but 'The lion rushed' would be metaphor ['lion' referring to a human hero]; since both are brave."

Metaphors are often used for expressing emotions, as a tool for visualizing concepts. A *broken heart* describes a sad feeling caused by someone or something; it is not meant literally. It creates an image of a heart that is broken into pieces for conveying an extreme feeling of sadness. In [10], it was shown that metaphors carry significantly more emotions than do literal expressions. This is one of the reasons for metaphor being a useful device in creative expression. For example, it allows a writer to describe a concept that is difficult to explain directly through a creative emotional imagery. In [7], *image metaphors* are defined as metaphors that map conventional mental images onto other conventional images with similar characteristics, as for example, describing a politician as a "bulldozer". This opens up many possibilities for creativity in writing.

A specific metaphor sometimes has an ambiguous interpretation. For example, when we say *memory is a river*, both *fluid* and *long* might be considered acceptable interpretations [16]. It has been shown In experiments [13] that sentential context, too, may affect the meaning of the metaphor. The emotional characteristic of metaphor increases the level of ambiguity, as people might interpret emotions in multiple ways.

The rhetorician, I. A. Richards [15], decomposes a metaphor into two main components: the *tenor* and the *vehicle*. The tenor, or *topic*, is that which is being described by potential meanings, referred to as *properties*, of the vehicle. There are several metaphorical syntactic constructions. Similarly to other works on this topic, we focus on Noun-Noun constructions, that is, metaphors of the form *Noun* is [a] *Noun*; *time is money*, for example. The first noun is the topic and the second, the vehicle. This type of metaphor is known as *nominal*. Noun-Noun constructions may extend beyond two nouns. For example, Albert Einstein once said: "All religions, arts, and sciences are branches of the same tree", suggesting that the three topics are related.

The meaning of a metaphor may be related more to the topic, the vehicle, or to both in the same level. For example, when one says that *Joe is a chicken*, the meaning is usually described as being *afraid*, which is more closely related to the vehicle *chicken* than to the topic *Joe*. On the other hand, Bob Dylan said in an interview on 1965, "Chaos is a friend of mine", a metaphor that can be interpreted as something *chaotic*, which is more related to the topic.

We describe a system that is designed for interpreting nominal metaphors, given without context. Similarly to previous works, we exploit a large corpus of text documents for semantically describing words and properties using a mathematical device. We use a word-embedding representation for calculating similarity between a candidate interpretation and the topic and the vehicle, so as to rank candidates based on a semantic score. As a final step, we automatically cluster results and keep only the best interpretations out of each cluster.

To summarize this paper's contributions:

1. We provide a new and improved dataset.
2. We extend previous works in this field using a richer semantic model for interpreting metaphors, and obtain competitive results.
3. We show that clustering and filtering the results to leave only the best in each cluster improves performance.
4. We show that using word associations as interpretation candidates, combined with collocations, improves performance, as do topic interpretations.
5. We suggest some additional metrics for evaluation, such as mean reciprocal rank and mean average precision. And we use word senses (WordNet synonyms) for matching.

The next section cites some related work. Our contributions and the results of experiments are described in the following two sections. Some conclusions are drawn in the final section.

2 Related Work

Different tasks relate to automatic metaphor processing. One is about automatically identifying metaphor in running text, that is, tagging words as being part of a metaphor or not. Many studies handle this. For example, Turney et al. [19] automatically tag words in a given context as either *literal* or *metaphorical*, by training a supervised classifier. They focus on features that measure the level of abstractness of the word's context. They were able to show state-of-the-art performance on a dataset of adjective-noun metaphors (e.g. *sweet child*). For more information on metaphor identification, we refer the reader to [17], a recent review of metaphor processing systems.

The computational task in which we are more interested, is *interpretation*, interpreting a given metaphor. This very challenging task has garnered interest over the past few years. *Metaphor Magnet* [20], allows users to enter a metaphor or simile, potentially augmented with sentiment polarity (e.g. $+/-$); for example, *life is a +game*, including a plus sign for *game*, indicating a positive sentiment. Using sentiment this way allows users to provide some information about the context. To interpret a metaphor, the system expands the topic and vehicle with some corpus-based *stereotypes*, and then with the stereotype's properties. The properties that saliently occur with both, the topic and the vehicle, are returned as results. For Metaphor Magnet, a stereotype is a word that describes the topic/vehicle. The stereotypes and properties are discovered using Google n-grams, as it contains n-grams of the form "X is a Y" that help one understand how X is typically being described.

There are a few works that treat the text components as vectors of a higher dimension in a semantic space. This opens the possibility of using mathematical tools to calculate the similarity of two components, through measuring the distance between their corresponding vectors. Kintsch [5] uses Latent Semantic Analysis (LSA) [1] for modeling the vector space. They generate term vectors that highly correlate with both, the topic and the vehicle; correlation is measured by cosine similarity over the LSA vectors. Metaphor interpretation is represented by the centroid vector of the most similar terms, and it does not necessarily represent a real word. Terai and Nakagawa [18] use the same algorithm, over a slightly different semantic model. They use Probabilistic Latent Semantic Indexing (PLSI) [4], for finding potential properties, limiting to adjectives and verbs. We go down the same path, in the sense that we use a semantic model for calculating a score for the candidate properties. Similarly, we focus on adjectives and verbs as the only possible interpretations. Terai and Nakagawa [18] extended their process with a recurrent neural network trained over the properties and scores for finding the dynamic interaction between the properties.

The most relevant work for us is *Meta4meaning* [21], an interpretation system for nominal metaphors. This work uses an LSA along with two dimensionality reduction techniques, Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF). It only considers abstract words as candidate properties. The properties are ranked according to their association strength with both, topic and vehicle. It uses different aggregation methods for combining the

association scores of the topic and the vehicle. The system shows a strong performance advantage over the human-annotated dataset provided by [16] compared with other systems.

Following *Meta4meaning*, we build a word-embedding model instead of LSA. Specifically, we use a 300-dimensional GloVe model [14]. Word embeddings, specifically of the type that we are using, outperform SVD for analogy tasks [9]. Since our task is more similar to analogy than to word similarity, we were led to believe that word embeddings may improve performance of metaphor interpretation.

3 Metaphor Processing

Given a metaphor, we begin by generating a list of interpretation candidates. We do this by finding collocations of the topic and vehicle individually, and consider each one of them as a potential candidate. For each candidate c, we calculate a topic semantic score, which is the cosine similarity between c and the k most significant collocations of the topic (k is a parameter) and aggregate it into a single score. We use average for aggregation. Similarly, we calculate a vehicle semantic score.

In the next step, we calculate two pointwise mutual information (PMI) values, between c and the topic and vehicle respectively. We add the frequency of c as another score and combine all the five score functions in a log-linear structure, with weights assigned to each. The weights are adjusted automatically, as we describe in the following section.

To remove semantically related interpretations from the list, we cluster the results and keep only the highest ranked candidates in each cluster. The remaining candidates are ranked according to their final score and the best n candidates (n, too, is a parameter) are returned as interpretations.

We now describe each step in greater detail.

3.1 Potential Interpretations

In our work, similar to other relevant works, e.g. [21,16], a metaphor *interpretation* is composed of a single word that conveys the main concept of the metaphor. For example, among the interpretations of the metaphor *city is a jungle* one can find *crazy* and *crowded*. It is natural to assume that an interpretation should be of a class of *describing* words, that is, words that are used for describing objects. Therefore, similar to other related works [20,21], we consider all adjectives as potential interpretations. Additionally, verbs with an *ing* ending are good candidates too. In [21], they only consider abstract words as potential interpretations. The level of abstractness of a word was measured by Turney et al. [19] automatically for about 11,000 words. To avoid the limitation in using such a list, we did not go that route; we believe that most of the potential interpretations are adjectives.

3.2 Dependency-Based Collocations

Our interpretation process begins with extracting collocations of the vehicle and the topic individually using a relatively large corpus. Specifically, we use DepCC,[1] a dependency-parsed "web-scale corpus" based on CommonCrawl.[2] There are 365 million documents in the corpus, comprising about 252B tokens. Among other preprocessing steps, every sentence was given with word dependencies discovered by MaltParser [12]. We only use a fraction of the corpus containing some 1.7B tokens. Here, we considered as collocation words that are found to be dependent in either the topic or the vehicle, and assigned with a relevant part-of-speech tag: adjective or verb+*ing*. The main assumption is that many potential modifiers of a given noun will appear somewhere in the corpus as a dependent in the dependency graph.

For example, the dependency-based collocations for *school* are: *high*, *elementary*, *old*, *grad*, *middle*, *med*, *private*, *attending*, *graduating*, *secondary*, *leaving*, and *primary*.

To eliminate noisy results that might transpire given that the corpus was generated from the open web, we preserve only candidates that have an entry in WordNet [2].

3.3 Word Association

In parallel with our objective data-driven collocation extraction process, we experimented with word associations as an alternative, more subjective, process for generating interpretation candidates. Word-association norms are repositories of pairs of words and their association frequency in a given population. The first word is a cue or trigger given to participants, and the second is the reported associated word that first came to a subject's mind. For example, *bank* is paired with *money*, because the cue *bank* often elicits the response *money*. Those pairs form various semantic-relation types; some might not be deemed symmetric. Word association norms have been used in psychological and medical research, as well as a device for measuring creativity.

We use the University of South Florida Free (USF) Association Norms [11][3] for generating alternative candidates. This repository contains 5,019 cue words that were given to 6,000 participants beginning in 1973. The way we work with this repository is by adding all the associated words of the topic and vehicle individually. In this case, we allow words of all POS classes to be considered as candidates. For example, the associations for cue *school* are: *work*, *college*, *book*, *bus*, *learn*, *study*, *student*, *homework*, *teacher*, *class*, *education*, *USF* (!), *hard*, *boring*, *child*, *house*, *day*, *elementary*, *friend*, *grade*, *time*, *yard*. We evaluate our system's performance with and without the associations; results are reported below.

[1] `https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/depcc.html`
[2] `http://commoncrawl.org`
[3] `http://w3.usf.edu/FreeAssociation`

3.4 Calculating Semantic Scores

For each candidate we calculate a couple of semantic scores, one for the topic and one for the vehicle. We use word embeddings to transform every word into a continuous vector that captures the meaning of the word, as evidenced in the underlying corpus. We used pre-trained GloVe [14] vectors; specifically, we use the ones that were trained over a 6B token corpus, comprising 400K vectors, each of 300 dimensions. In what follows, we denote the vector of a word v by w_v.

We believe that the most significant collocations of the topic/vehicle tend to reliably represent the way the topic/vehicle, respectively, can be described in different contexts. Therefore, the semantic score $sem(c,t)$ of a candidate c and the topic t is the average cosine similarity between w_c and the vectors of the k most significant collocations of t. Similarly, $sem(c,v)$ is the semantic score of a candidate c and the vehicle v. We experimented with different values for k. Results are reported in the next section.

3.5 Final Scores

For each candidate c, we calculate $npmi(c,t)$ and $npmi(c,v)$, the normalized Pointwise Mutual Information (PMI) values for the topic and vehicle, respectively. Normalized PMI is similar to PMI, except that it is normalized between -1 and 1. The PMI between a candidate c and a noun n is calculated over the dependency graph; that is, we calculate the chances of seeing c as a dependent of n in a dependency graph. We add $freq(c)$, the frequency of c as another score, calculated over the entire corpus.

To summarize, given a candidate c, the full list of scores is

$$\langle sem(c,t), sem(c,v), npmi(c,t), npmi(c,v), freq(c)\rangle$$

combined using a log-linear structure, with each score amplified by a weight:

$$FinalScore(c) = \sum_{k=1}^{5} \lambda_k \log score_k$$

We automatically adjust these weights over a development set of metaphors and interpretations to optimize for recall, as explained below. As a result, each candidate is ranked according to its final score.

3.6 Clustering

Lastly, we cluster the list of candidates as a way to deduplicate it. We run clustering using word vectors for finding groups of words that have a strong semantic association of any kind, keeping only the best candidates in each cluster.

We use Density-Based Spatial Clustering of Applications with Noise (DBSCAN) for clustering. This method groups together vectors that are bundled in

Table 1. Results for several metaphors.

"friendship is a rainbow"	"god is a fire"	"typewriter is a dinosaur"
beautiful	*burning*	*prehistoric*
wonderful	*fighting*	*fossilised*
colorful	*holy*	*extinct*
forming	*sacred*	*resembling*
pink	*good*	*feathered*
great	*absolute*	*robotic*
bright	*powerful*	*stuffed*
magical	*cannon*	*primitive*
deep	*dangerous*	*preserved*
double	*killing*	*gigantic*
happy	*almighty*	*antique*
featuring	*calling*	*lumbering*
good	*great*	*basal*
vibrant	*heavy*	*ancient*
glorious	*alive*	*oversized*

the space by forcing a minimum number of neighbors. Vectors that do not have the requisite number of neighbors, or in other words occur in low-density areas, are reported as noise and are not placed under any cluster. For us it means that they were not connected with other vectors, so they might have a unique meaning among the listed candidates. We treat such vectors as if they form singletons.

For example, among the interpretation candidates for the metaphor *anger is fire* we find *red* and *black*. After clustering, *black* is removed. As another example, the following candidates for the metaphor *a desert is an oven* may be grouped together: *eating*, *healthy*, *delicious*, *fried*, *spicy*, *leftover*, *veggie*, *steamed*, *lentil*, *roasted*, *homemade*, *yummy*, *creamy*, *glazed*, *seasoning*, *crunchy*, *baking*. (These likely result from the frequent misspelling of "dessert" in the corpus used.)

There are two parameters that need to be configured for DBSCAN: (1) ε – the radius of the consideration area around every vector; and (2) μ – the minimum number of neighbors required in the consideration area. The distance measure should also be configured. We use the common Euclidean distance, which usually shows good performance in a relatively low-dimension space like ours. Below we describe our experimental results, using different values for both parameters. Table 1 shows a few outputs for three different metaphors.

4 Experimental Results

4.1 Evaluation Set

We evaluate our system with the dataset published by [16], containing 84 unique topic/vehicle pairs that were associated with interpretations by twenty different

Table 2. Topic/vehicle pairs and associated properties.

Topic/Vehicle pair	Associated properties
Skating/Flying	*Free*; *Fast*; *Relaxing*
Store/Zoo	*Crowded*
Wisdom/Ocean	*Vast*; *Huge*
Job/Jail	*Boring*

study participants. Each participant was asked to assign interpretation for different aspects of the pairs, such as treating a pair as a metaphor (e.g. *knowledge/power*, from the phrase *knowledge is power*) or as a simile (e.g. *knowledge/power*, from *knowledge is like power*). We focus on the interpretation of metaphors, both lexicalized and non-lexicalized.

As a preprocessing step, we lemmatize the interpretations, so as to allow our method's results and the true interpretations to match more smoothly. Additionally, we allow interpretations to match if they are considered as synonyms in WordNet. In this work we focus on nominal metaphors, and since our collocation as well as word-embedding models were trained to handle unigrams, we had to modify some of the metaphors that have multiword vehicles; for example *sermon is a sleeping pill*. Therefore, we modified such multiword vehicles into one word simply by eliminating the space characters, knowing it may cause performance reduction; for example, *sermon is a sleeping pill* was modified to *sermon is a sleepingpill*.

Each metaphor might be associated with more than one interpretation. As do other related works [21,16], we only consider interpretations that were assigned by at least five participants; we call them *qualified interpretations*. This leaves us with only 76 qualified metaphors (i.e. metaphors with at least one qualified interpretation), with two qualified interpretations per metaphor on average. Table 2 shows a few examples of interpretations as assigned by 20 human annotators for the dataset of [16].

4.2 Evaluation Method

To stay in line with related works [21], we report *Recall* @K, which is the average percentage of human-associated interpretations that are found in the top K results. For example, the following results were generated for the pair *skating/flying* from Table 2: *incredible*, *high*, *free*, *great*, *fast*. Therefore, Recall@3 is 33%, while Recall@5 is 66%. We compare our results with [21], which was evaluated on the same dataset following a similar preprocessing step. Therefore, we report on Recall at their reported K's: 5, 10, 15, 25, and 50.

To measure the false positives reported by the system, we evaluate the results with two additional standard metrics: mean reciprocal rank (MRR) and mean average precision (MAP).

Table 3. System parameters tuned to maximize MRR, MAP and Recall@K. The second column shows the range of values considered.

Param	**Range**	**MRR**	**MAP**	**@5**	**@10**	**@15**	**@25**	**@50**
DBSCAN ε	1 .. 6	4	5	4	5	5	4	4
DBSCAN μ	1 .. 5	4	1	6	1	5	5	5
DBSCAN n	1 .. 12	1	1	1	1	1	1	1
$sem(c,t)$	0.1 .. 1	0.6	0.6	0.6	0.1	0.1	0.6	0.6
$sem(c,v)$	0.1 .. 1	1.1	1.1	1.1	0.6	0.6	1.1	1.1
$npmi(c,t)$	0.1 .. 1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
$npmi(c,v)$	0.1 .. 1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
$freq(c)$	1 .. 10	3	3	5	7	7	5	3

4.3 Tuning System Weights

Our log-linear structure is composed of a set of weighted score functions. We adjust the scores using a tuning process over a development set, composed of about 50% of the metaphors. For each weight, we explore a range of possible scores, while we test all possible score combinations taking the brute force approach. For all scores except *freq*, we consider the range 0.1 .. 1; because of scale differences, for *freq* we consider the range 1 .. 10.

As mentioned, we use DBSCAN to cluster the list of candidates so as to remove some of the semantically related ones. We take a similar brute force approach for tuning the DBSCAN parameters, ε and μ. We also tune n, the number of top results taken from each cluster. For tuning, we use the same development set, evaluated over MRR, MAP and Recall @K values. Table 3 shows the ranges and best values of all the parameters we tune.

We see that both semantic scores get higher weights than the npmi scores, suggesting that the semantic distance as measured by cosine similarity between the vectors of the candidates and the collocations of the topic/vehicle, is effective. The DBSCAN parameters are less stable across different metric optimizations. One thing we learn is that when optimizing for larger values of @K, DBSCAN requires dense areas around clustered vectors, resulting in a lower number of clusters. Additionally, the system does not benefit from high values of the DBSCAN n parameter. It turns out that it is better to consider only one interpretation from each cluster.

4.4 Evaluation Results

We evaluate our system against the 76 "qualified" metaphors in the dataset. For each metaphor, our system generates the top 100 interpretation results, which are then compared with the metaphor's human-associated qualified interpretations. For the clustering parameters and scoring weights, we use the tuned values

Table 4. Each row shows evaluation results when using optimal parameter values for the metric mentioned in the first column.

Optimization	**MRR**	**MAP**	**@5**	**@10**	**@15**	**@25**	**@50**
MRR	**0.312**	**0.170**	0.198	0.254	0.278	0.405	0.562
MAP	**0.312**	**0.170**	0.198	0.254	0.278	0.405	0.562
@5	0.302	0.166	**0.207**	0.258	0.270	0.430	0.548
@10	0.233	0.151	0.180	**0.273**	0.322	0.374	0.521
@15	0.245	0.160	0.151	0.262	**0.331**	0.392	0.513
@25	0.302	0.166	**0.207**	0.258	0.270	**0.430**	0.548
@50	**0.312**	**0.170**	0.198	0.254	0.278	0.405	**0.562**

Table 5. Comparison with Meta4meaning.

System	**MRR**	**MAP**	**@5**	**@10**	**@15**	**@25**	**@50**
Meta4meaning	N/A	N/A	**0.221**	**0.303**	**0.339**	0.397	0.454
Ours	0.312	0.170	0.198	0.254	0.278	**0.405**	**0.562**

reported in the previous subsections. Since we tune for different evaluation metrics, here we individually use each set of values for generating the top 100 results and calculating MRR, MAP and recall at all the relevant K values. Table 4 summarizes the evaluation results at MRR, MAP and Recall@5, @10, @15, @25, and @50, for each set of parameter values. We observe that when optimizing the system for Recall@50 we at least get close to the best result for all other evaluation metrics. Therefore, in what follows we use the parameter values optimized for @50.

We compare our results with the ones reported by Meta4meaning [21], evaluating over the same set of metaphors and following similar preprocessing steps. Table 5 compares the results reported by both systems. While our system somewhat underperforms for the lower values of Recall @k, it is doing slightly better on @25 and @50. These results show that, while our system has a better overall coverage, correct interpretations are concentrated more in the lower part of the ranked list that we produce. With more work, we expect to be able to filter out many of the non-associated interpretations, thereby ranking the correct ones higher in the list.

To measure the effect of clustering on the results, we evaluate our system running with and without clustering. When running with clustering, we use the optimized set of parameters, as reported in Table 3. Table 6 compares our system's results, with and without clustering. We learn that when using clustering, our system was able to eliminate noise in lower parts of the ranked list of candidates, thereby making room for alternative and correct interpretations that ranked lower without clustering.

Table 6. Evaluation results, with and without clustering.

Method	@5	@10	@15	@25	@50
w/o clustering	0.198	0.254	0.278	0.351	0.534
w/ clustering	0.198	0.254	0.278	**0.405**	**0.562**

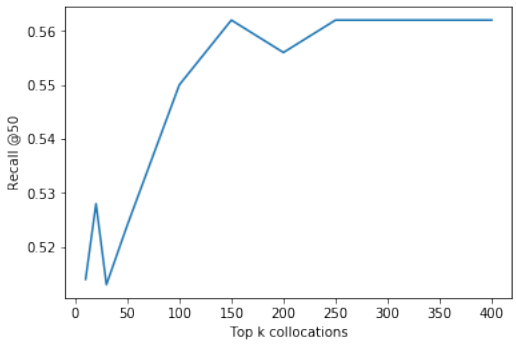

Fig. 1. Evaluation results as Recall@50, measured over different k values for the maximum number of collocations we take from the topic/vehicle for calculating the semantic score.

Recall that our topic/vehicle semantic scores are defined as the cosine similarity between the candidate vector and the top k collocations of the topic/vehicle. We tested our system with different k values; Figure 1 shows evaluation results as Recall@50 when running the system with different k values. Observe that it gets maximized at higher values of k, suggesting that the meaning of the topic/vehicle is usually more complex, and that it takes multiple properties to describe when comparing it vis-à-vis candidate interpretations.

Finally, we check how our system's performance is affected by adding word associations as an additional source for generating interpretation candidates. When we run our system using only dependency-based collocations as candidates, we obtain Recall@50 score of 0.551. This was improved to 0.562 when we add word associations as candidates.

4.5 Improving the dataset

Overall, the system could not generate even one correct interpretation (among its 50 best results) for 20 out of the 76 evaluated metaphors. Some of those metaphors did not have a correct interpretation anywhere in the list, even beyond the best 50; for example, *music is a medicine*. Taking a closer look at the dataset, we found that some metaphors did not come with any correct interpretation in its interpretation list, even when taking into account all the provided

Table 7. Evaluation results when running on different datasets.

Dataset	MRR	MAP	@5	@10	@15	@25	@50
Original	0.312	0.170	0.198	0.254	0.278	0.405	0.562
Improved	0.151	0.073	0.051	0.070	0.114	0.171	0.311

interpretations, not just qualified ones. For example, take the metaphor *education is a stairway*. The suggested interpretations are *higher*, *steps*, *upward*, *long*, *passage*, *ascension*, *climbing* – none of which qualified. Most of these interpretations do not reflect the true meaning of this metaphor (*steps*, *passage* and *climbing* are themselves metaphors; *long* is surely not intended; *higher* and *upward* make little sense); we would rather suggest *enabling* as a more suitable interpretation. For *job is a jail*, the only qualified interpretation is *boring*, while the more accurate interpretation, *confining*, was proposed by fewer then 5 annotators, and therefore did not pass the bar. These are only a few of the examples that encouraged us to perform our own annotation process over the entire dataset. This was done by a native English speaker. We override the original interpretations with the newer ones, resulting in a slightly larger dataset, because with the new annotations some unqualified metaphors now qualify.

In addition to these new annotations, we extended the dataset with 14 new metaphors extracted from [8], among them *words are weapons* and *logic is gravity*. We followed the same annotation process to assign interpretations for the new metaphors. The extended (and improved) dataset contains 98 metaphors with refined interpretations. The full list of modifications can be found in the dataset (published at *to be supplied in the final version*). We intend to extend it even further in the future.

Table 1 compares evaluation results for the original and improved datasets. The degraded results we get for the latter is explained by the fact that, for most metaphors in the dataset, our improvement process removed the majority of suggested interpretations. Fewer human-annotated interpretations means fewer successful matches, making our improved dataset harder to interpret to begin with.

5 Conclusions

We have described a system that interprets nominal metaphors, provided without a context. Given a metaphor, we generate a set of interpretation candidates and rank them according to how strongly they are associated with the topic, as well as with the vehicle. Candidates are generated using two techniques. First, we find collocations of the topic and vehicle, focusing on adjectives as well as gerunds, which were found to be dependent of the topic/vehicle in at least one sentence in a large corpus. We add to that list word associations of both. This addition has proven effective.

Our ranking procedure combines a number of scores assigned for each candidate, which are based on normalized PMI as well as cosine similarity between the representing GloVe vectors of the candidates and the topic/vehicle collocations. The scores are aggregated using a weighted log-linear structure. We tune the weights automatically, optimizing for various evaluation metrics: MRR, MAP and Recall@K for different K values. We found that with small K, the similarity between candidate and topic becomes more important than other score functions. overl In a post-processing step, we cluster the results using DBSCAN and keep only the best candidates out of each cluster. Our system benefits thereby.

Our system was evaluated against a set of metaphors that were assigned with properties by 20 human evaluators. We compare our results with Meta4meaning and obtained competitive results.

Additional work is needed to handle the cases mentioned in the analysis section, especially, cleaning the results from candidates that have an opposite meaning from the ones we are looking for.

Potential future directions include working on additional types of metaphors, as well as additional languages. We plan to improve the current evaluation technique; one option, which we're considering, is to measure the effect of metaphor interpretation on common NLP tasks, such as machine translation. We will also be looking at the analysis of metaphors in context.

References

1. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6), 391–407 (1990)
2. Fellbaum, C. (ed.): WordNet: an electronic lexical database. MIT Press (1998)
3. Gentner, D., Bowdle, B.F., Wolff, P., Boronat, C.: Metaphor is like analogy. In: The Analogical Mind: Perspectives from cognitive science, chap. 6, pp. 199–253. MIT Press, Cambridge, MA (2001)
4. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 50–57. ACM (1999)
5. Kintsch, W.: Metaphor comprehension: A computational theory. Psychonomic Bulletin & Review 7(2), 257–266 (2000)
6. Krennmayr, T.: What corpus linguistics can tell us about metaphor use in newspaper texts. Journalism Studies 16(4), 530–546 (2015)
7. Lakoff, G.: Image metaphors. Metaphor and Symbolic Activity 2(3), 219–222 (1987)
8. Lakoff, G., Espenson, J., Schwartz, A.: The master metaphor list. Tech. rep., University of California at Berkeley (1991)
9. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in Neural Information Processing Systems (NIPS). pp. 2177–2185 (2014)
10. Mohammad, S.M., Shutova, E., Turney, P.D.: Metaphor as a medium for emotion: An empirical study. In: Proceedings of the Joint Conference on Lexical and Computational Semantics (*Sem). p. 23 (2016)

11. Nelson, D.L., McEvoy, C.L., Schreiber, T.A.: The University of South Florida free association, rhyme, and word fragment norms. Behavior Research Methods, Instruments, & Computers 36(3), 402–407 (2004)
12. Nivre, J., Hall, J.: Maltparser: A language-independent system for data-driven dependency parsing. In: In Proc. of the Fourth Workshop on Treebanks and Linguistic Theories. pp. 13–95 (2005)
13. Ortony, A., Schallert, D.L., Reynolds, R.E., Antos, S.J.: Interpreting metaphors and idioms: Some effects of context on comprehension. Journal of Verbal Learning and Verbal Behavior 17(4), 465–477 (1978)
14. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014), http://www.aclweb.org/anthology/D14-1162
15. Richards, I.A.: The Philosophy of Rhetoric. Bryn Mawr College. Mary Flexner lectures, Oxford University Press (1936; reprinted 1965)
16. Roncero, C., de Almeida, R.G.: Semantic properties, aptness, familiarity, conventionality, and interpretive diversity scores for 84 metaphors and similes. Behavior Research Methods 47(3), 800–812 (2015)
17. Shutova, E.: Design and evaluation of metaphor processing systems. Computational Linguistics 41(4), 579–623 (Dec 2015)
18. Terai, A., Nakagawa, M.: A corpus-based computational model of metaphor understanding consisting of two processes. Cognitive Systems Research 19, 30–38 (2012)
19. Turney, P.D., Neuman, Y., Assaf, D., Cohen, Y.: Literal and metaphorical sense identification through concrete and abstract context. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 680–690. EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA (2011), http://dl.acm.org/citation.cfm?id=2145432.2145511
20. Veale, T., Li, G.: Specifying viewpoint and information need with affective metaphors: A system demonstration of the metaphor magnet web app/service. In: Proceedings of the ACL 2012 System Demonstrations. pp. 7–12. ACL '12, Association for Computational Linguistics, Stroudsburg, PA (2012), http://dl.acm.org/citation.cfm?id=2390470.2390472
21. Xiao, P., Alnajjar, K., Granroth-Wilding, M., Agres, K., Toivonen, H.: Meta4meaning: Automatic metaphor interpretation using corpus-derived word associations. In: Proceedings of the Seventh International Conference on Computational Creativity (ICCC). pp. 230–237. Paris, France (June 2016)