A model for high-coverage lexical semantic annotation generation

Attila Novák, Borbála Siklósi
Pázmány Péter Catholic University Faculty of Information Technology and Bionics,
MTA-PPKE Hungarian Language Technology Research Group
1083 Práter u. 50/a
Budapest, Hungary

Abstract

AI applications often receive their input in the form of natural language text, or as the transcription of spoken text. A commonsense inference system should transform such input to a formal representation with limited vocabulary in order to be able to process them. In this paper, we present a method based on neural word embeddings that automatically assigns semic features to words of natural language. These features either describe the ontological category of a given word or provide some characterization or additional information. We show that our method has high coverage and performs well for English and Hungarian, and can easily be extended to other languages as well.

Introduction

One of the most natural representations of commonsense knowledge is natural language. What people think or know about the world is expressed in either spoken or written language. Due to the popularity and accessibility of on-line media, crowds of people put their knowledge into written texts, either in the form of very short comments on social media sites or in the form of longer posts in addition to the writings of professional journalists. These texts, which are produced in a daily manner, adapt to changes in language use, and not only general knowledge, but facts and beliefs about the actual state of the world is also represented in them. Moreover, not only standard language, but slang and words used in informal contexts and special domains are also present in texts collected from the Web. In addition, more and more books representing a wide range of domains and styles are digitized. Large written corpora consisting of these resources are available as raw material for research, and can be exploited as a source of knowledge.

A more structured form of knowledge representation is hand-crafted ontologies, such as WordNet (Fellbaum 1998; Miller 1995) or DBpedia (Lehmann et al. 2015). In WordNet, concepts are collected into synonym sets and are organized into a strictly hierarchical structure of hyponymy relations, along with some horizontal relations, like meronymy. However, WordNet has been criticized for its too high granularity at the bottom level and its generality at the top level (Brown 2008). Moreover, its middle layers also contain many concepts that may be appropriate in a scientific taxonomy, like 'fissiped mammal.n', but are not present in everyday language use. Similar problems concern most other structured knowledge bases. Moreover, since they are extremely costly to produce or extend to achieve a good lexical coverage, these resources are static in nature, they are not able to keep up with changes in language use and daily life, and they contain only standard word forms.

Whatever its source, a knowledge base is an essential component of a commonsense inference system. Even though recent results achieved by applying deep neural systems on raw textual input have been significant, traditional inference systems first transform their input written in natural language into a formal representation using features extracted from one or more knowledge bases, then they try to solve the given task based on this formal representation. In order to be able to process arbitrary input, the coverage of the knowledge bases used should be as high as possible (Davis 1990).

In this paper, we present an automatic method that is able to assign semantic features or atomic predicates to practically any (even non-standard/slang or misspelled) word form in a text in a language-independent manner. As we apply morphological analysis and lemmatization to the corpus both at the time of generating the embedding models and at query time, all forms of a single lemma are covered instead of only those explicitly present in the original corpus. This is essential to achieve a good coverage for an agglutinating language like Hungarian where a single lexeme may have hundreds of possible word forms, only few of which are actually present even in a huge corpus. Instead of constructing another static knowledge base of fixed vocabulary, we propose a dynamic tool that can be retrained or fine-tuned at any time using an up-to-date, possibly domain-specific corpus appropriate to the task at hand. The target formalism or set of semantic features to be used is also an interchangeable parameter of the proposed method. The set of features and predicates presented in this paper is derived from formalized definitions of a subset of the headwords (including the defining vocabulary) of the Longman Dictionary of Contemporary English (LDOCE) (Summers 2005). Both the vocabulary of the model and the features used are embedded in a neural-network-created word embedding vector space model (Mikolov et al. 2013).

Before we present the structure of the paper, let the fol-

lowing example illustrate the kind of semantic annotation automatically assigned by the model to words in the sentence *The cow gives milk to her calf.*:

```
cow: mammal, at_farm, produce_milk, HAS{four(legs)}, animal
gives: =AGT.CAUSE{=DAT.HAS.=PAT}, give, offer, communicate
milk: food, sweet, drink, liquid
calf: young, mammal, animal, has_wool. HAS{four(legs)}
```

The paper is structured as follows: first, a brief introduction to neural word embeddings is presented. This is followed by the description of the lexical resource that we used when creating our models. In the following section, the method of building the model is described. In this paper, the method is demonstrated for English. However, existing semantic resources can also be mapped to word embedding spaces over the vocabulary of other languages. We have performed experiments with Hungarian, an agglutinative language with scarce semantic resources, but the method can easily be applied to other languages as well. Finally, we present both qualitative and quantitative evaluation of the models.

Word Embedding Models

Traditional models of distributional semantics build word representations by counting words occurring in a fixed-size context of the target word (Baroni, Dinu, and Kruszewski 2014). In contrast, more recent methods for building distributional representations of words use neural networks to generate *word embedding models* (Mikolov et al. 2013; Pennington, Socher, and Manning 2014) the most influential implementation of which is word2vec[1].

When training embedding models, a fixed-size context of each word in the vocabulary is used as the input of a neural network. This network is used to predict the target word from the context by using back-propagation and adjusting the weights assigned to the connection between the input neurons (each corresponding to an item in the whole vocabulary) and the projection layer of the network. This weight vector can finally be extracted and used as the embedding vector of the target word. Since similar words are used in similar contexts, these vectors optimized for prediction from the context will also be similar for similar words. There are two types of neural networks used for this task. One of them is the so called CBOW (continuous bag-of-words) model in which the network is used to predict the target word from the context, while the other model, called skip-gram, is used to predict the context from the target word. For both models, the embedding vectors can be extracted from the middle layer of the network and can be used alike as a dense vector representation of the meaning of the words in both cases.

The vectors thus obtained point to certain locations in the semantic space consistently so that semantically and/or syntactically related words are close to each other, while unrelated ones are more distant. Moreover, it has been shown that vector operations can also be applied to these representations, thus the semantic relatedness of two words can be quantified as the algebraic difference of the two vectors representing these words. Similarly, the meaning of the composition of two (or more) words is generally well represented by the sum of the corresponding embedding vectors (Mikolov, Yih, and Zweig 2013).

As the words are represented as dense real-valued vectors, the similarity of two words can easily be defined as the angle between the vectors of the words, i.e. the most similar words for a query word can be retrieved by finding its nearest neighbours in the vector space according to cosine distance.

One of the main drawbacks of building such a model from raw corpora, however, is that by itself it is not able to handle polysemy and homonymy, because one representational vector is built for one lexical element regardless of the number of its different senses. We applied a simple method to alleviate this problem, at least in cases where the homonyms have different PoS. In order to assign different vectors to the same word with different parts-of-speech, we applied PoS-tagging and lemmatization to the training corpora before building the model. The main PoS tag of each word was attached to the word as a suffix in the form *lemma#PoS*, thus a different embedding vector was created for homonymous lemmas with different parts-of-speech.

We trained an English word embedding model on the English Wikipedia dump[2] of 2.25 billion tokens (8.24 M token types) that was annotated using the Stanford tagger (Toutanova et al. 2003). Since the CBOW model has proved to be more efficient for large training corpora, we used this model architecture for training with the radius of the context window set to 5 and the number of dimensions to 300 and using a token frequency limit of 5.

Figure 1 illustrates how the words *pianist, teacher, turner, maid* and their three nearest neighbors are arranged in the English word embedding space[3]. The original vectors consist of 300 dimensions, but these were mapped to a 2D representation using the t-sne algorithm (van der Maaten and Hinton 2008).

Lexical Resources

Our goal was to create a model that can assign semantic features and elementary predicates to words in an arbitrary text. Thus, first, the set of features to be used had to be defined. The Longman Dictionary of Contemporary English (LDOCE) (Summers 2005) is a traditional dictionary containing words and their definitions. All definitions in the dictionary are written using a constrained defining vocabulary (Longman Defining Vocabulary (LDV)). The definitions of a subset of headwords in LDOCE, including all items in LDV and most frequent words listed in the BNC and the Google unigram count, were transformed into a formal description containing only unary and binary predicates in a resource called 4lang (Kornai et al. 2015). illustrated by the following examples (for the explanation of the notation used in these definitions see (Kornai et al. 2015)):

```
bread: food, FROM/2742 flour, bake MAKE
  (a type of food made from flour and water that is
```

[1] https://code.google.com/archive/p/word2vec/

[2] downloaded from https://dumps.wikimedia.org/ in May, 2016.

[3] The PoS tag is NN for all example words, and it is omitted from the figure.

Category	Example words in 4lang
PART_OF.body	body#NN, tongue#NN, back#NN, neck#NN, shoulder#NN, bone#NN, skin#NN, wrist#NN, buttock#NN etc.
=AGT.HAS.mouth	swallow#VB, suck#VB, eat#VB, drink#VB
HAS{four(legs)}	horse#NN, tiger#NN
mammal	mammal#NN, lion#NN, deer#NN, man#NN, horse#NN, sheep#NN, cattle#NN, rabbit#NN, cat#NN, pig#NN, goat#NN, cow#NN
=AGT.HAS.mind	read#VB, remember#VB, feel#VB, understand#VB
=AGT.CAUSE{=DAT.KNOW.=PAT}	express#VB, teach#VB

Table 1: Example words for some semantic features (predicates) after transforming the definitions to the format consisting of labels and example words

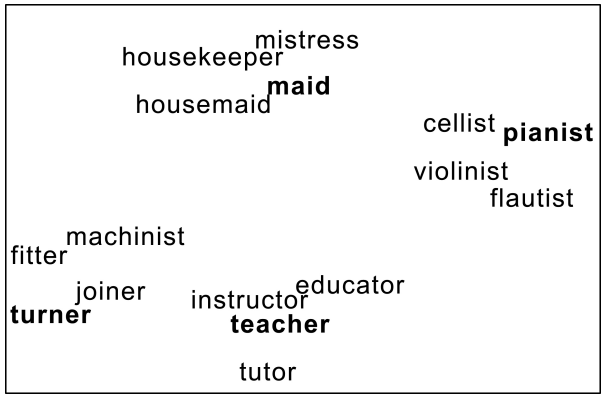

Figure 1: The arrangement of the 3 nearest neighbors of the words *pianist, teacher, turner, maid* in the English word embedding space

```
   mixed together and then baked)
show: =AGT CAUSE[=DAT LOOK =PAT], communicate
  (to let someone see something)
```

We further transformed this format so that we have some category labels (here: unary and binary predicates) and listed examples. This was achieved by segmenting the formal descriptions into elementary predicates (by splitting at commas), but we did not segment predicates into further parts, so e.g. HAS[four.(legs)] remained an atomic feature. Each such token was treated as a category label. Then, all words that had the particular token in their definition were listed as an example for that label. This resulted in 1489 category labels and 12,507 words listed as examples for them. Then, in order to make this resource compatible with the word embedding model built from the Wikipedia corpus, its vocabulary was intersected with that model. Even though the vocabulary of this resource consists mostly of frequent words used in LDOCE definitions, it also includes some affixes, inflected forms, and a few multiword items, which are not present in the lemmatized Wikipedia model, so the intersection resulted in 11,039 words. Table 1 shows some examples words for some features derived from the 4lang resource.

However, some categories were too broad and the set of words listed for them was too heterogeneous. To handle this problem, a hierarchical agglomerative clustering algorithm was applied to the set of words in those categories that contained at least five words. The reason for applying a hierarchical clustering rather than k-means is based on the argument of (Pereira, Tishby, and Lee 1993), who states that due to the sophisticated variability of written texts, the number of clusters of the concepts used in a certain text cannot be predicted. A hierarchical organization, however, is appropriate for producing compact groups of words and phrases, based on the actual text, rather than on some predefined generalization. The linkage method for the hierarchical clustering was chosen based on the cophenet correlation between the original data points and the resulting linkage matrix (Sokal and Rohlf 1962). The best correlation was achieved when using Wards distance criteria (Ward 1963), resulting in small and dense groups of terms at the lower level of the resulting dendrogram. However, we did not need the whole hierarchy, represented as a binary tree, but separate, compact groups of terms, i.e. well-separated subtrees of the dendrogram. The most intuitive way of defining these cutting points of the tree is to find large jumps in the clustering levels. To put it more formally, the height of each link in the cluster tree is to be compared with the heights of neighbouring links below it in a certain depth. If this difference is larger than a predefined threshold value (i.e. the link is inconsistent), then the link is a cutting point. For more details of the clustering algorithm, see (Siklósi 2016). Each cluster was then labeled with the original category label with a numeric index added.

Even though we present our method using only the 4lang dictionary as a lexical resource, the system can be built from any dictionary that can be transformed to a similar format.

Method

Our objective was to create a model with high lexical coverage that can also return the most relevant semantic features for words not present in 4lang. In order to achieve this goal, the semantic features from this controlled set were projected into the embedding space containing the representation of the words. Nearest feature neighbors for each word can be retrieved from the model using the cosine distance metric.

For each indexed semantic predicate label output by the clustering algorithm, we iterated the list of example words annotated with their part-of-speech (the crude PoS tags used in the 4lang resource had to be mapped to the more fine-grained PTB tags returned by the Stanford tagger) and retrieved their embedding vectors from the word embedding model built from the PoS-tagged Wikipedia corpus. As a simple but effective method for rendering a representation vector for a set of words with their corresponding word embeddings we took the mean of these vectors, and used that as the embedding vector of that particular semantic feature.

Original word	Analyzed word	Features
Laika	Laika#NNP	carnivorous mammal faithful HAS.short(hair/3359) HAS{four(legs)} ⟨AT/2744.farm⟩ companion young EAT.flesh HAS.long(tail)
likes	like#VB	want =PAT{person} wish emotion ask =AGT.HAS.mind annoy =PAT.IN/2758.mind communicate desire =AGT.HAS.body
eating	eat#VB	swallow =AGT.HAS.mouth eat love INSTRUMENT.tongue =AGT.CAUSE{=PAT{move}} sleep suck sing touch rest
fried	fried#JJ	food '.COOK/825 '.SERVE thick/2134 FROM/2742.flour bake.MAKE FROM/2742.milk food.IN/2758 vegetable sweet bread
onion	onion#NN	'.COOK/825 vegetable fruit food FROM/2742.milk sweet round soft thick/2134 PART_OF.plant
with	with#IN	
cucumber	cucumber#NN	vegetable fruit food '.COOK/825 sweet '.EAT round CAUSE{food.HAS.taste} PART_OF.plant soft

Table 2: An example sentence, *Laika likes eating fried onion with cucumber* with features assigned to each word using our method

Original word	Analyzed word	Hypernyms
Laika	Laika#NNP	
likes	like#VB	desire want
eating	eat#VB	consume digest take_in take have
fried	fried#JJ	
onion	onion#NN	vegetable produce food solid matter physical_entity entity
with	with#IN	
cucumber	cucumber#NN	vegetable produce food solid matter physical_entity entity

Table 3: An example sentence, *Laika likes eating fried onion with cucumber* with hypernyms from WordNet assigned to each word

Thus a representation of each predicate used in the definitions was obtained in the semantic space created from the English PoS-tagged corpus. These semantic feature vectors were kept separated from the word vectors in the original embedding model in order to be able to restrict lookup to either words or features derived from each lexical resource. To find the relevant features for a query word tagged with its appropriate part-of-speech, its representational vector is retrieved from the word embedding model and its nearest neighbors are taken from the model containing the semantic predicates. Since instead of exact matching, nearest neighbors are searched for, out-of-vocabulary words (with respect to the original lexical resources) can also be assigned semantic labels. The only requirement is that the word must be present in the word embedding model.

Other languages

We also carried out some experiments to apply our method to another language, Hungarian. Hungarian is an agglutinative language with very few lexical semantic resources. As the original 4lang dictionary contained the Hungarian translation of the vocabulary included (3477 words), it was straightforward to create a similar model for Hungarian as well. For this, we had to create a Hungarian word embedding model, which was built from a web-crawled corpus of 3.18 billion tokens (27.49 M token types) that was annotated using the PurePos (Orosz and Novák 2013) tagger, augmented with the Humor Hungarian morphological analyzer (Novák 2014; Novák, Siklósi, and Oravecz 2016). We applied the method described above to define the position of the features in the Hungarian word embedding space by calculating the mean of the vector representations of the Hungarian example words for each semantic predicate. Our approach can easily be extended to any other language by translating this dictionary of moderate size (relative to complicated knowledge bases). Furthermore, this method also adapts to differences in word usage in different languages, since words are represented with their embedding vector in the target language.

Experiments and Results

The aim of this research was to investigate the possibility of providing a high coverage tool for assigning a semantic representation to words of a natural language input dynamically instead of using a static knowledge base with a limited vocabulary. Thus, first we investigated the performance of the tool for some example input, then we also performed a quantitative analysis.

Qualitative analysis

Table 2 shows an example: *Laika likes eating fried onion with cucumber.* First, using the Stanford parser, the input is annotated with part-of-speech tags and each word is lemmatized. Then, for each lemmatized content word (i.e. omitting the function word *with*) with corresponding part-of-speech, the top 10 nearest features are retrieved from the model and ordered by their distance from the vector representing the target word in the embedding space. Note that the number of top n features generated for each word is a free parameter, but moving further in the semantic space results in less and less appropriate features for the target word. Table 3 shows the WordNet hypernyms assigned to each content word in the same sentence (the representation of the adjective *fried* and the proper name *Laika* is missing from WordNet).

As it can be seen in the example, our model is able to assign two types of features to words. Ontological/taxonomic categories, such as *carnivorous, mammal* for the word *Laika vegetable, food* for the words *onion* and *cucumber* appear together with characteristic features of the given concept, such as *faithful, HAS{four(legs)}, ⟨AT/2744.farm⟩* or *round*

and *CAUSE*{*food.HAS.taste*}. While the first type of features can be extracted from traditional ontologies, the latter type of characteristics can not. However, we believe that the latter type of features form an important part to common sense knowledge, because if people are asked to describe a concept, they will rather use such characteristics. Moreover, an inference system can also benefit from such descriptions. It can also be seen from the example, that the model "knows" that Laika is a dog by returning semantic features characterizing dogs. In addition, the feature *EAT.flesh* emphasizes the contrast of Laika being a dog and eating cucumber and onion.

Another benefit of our model, as mentioned above, is that it is able to generate features for all the words that are present in the original corpus the word embedding was built from, not only for the extremely limited set of words included in the 4lang dictionary. WordNet or other hand-made resources are limited only to the words and the classification that the designer of the resource had in mind. Our model, in contrast, is able to assign features to proper names, slang words or mistyped word forms as well as long as these are represented in the corpus the word embedding model was created from. In addition to the above example containing the dog name *Laika*, the following examples show some of the nearest features for two more proper names and two slang words:

IBM: information.IN, computer, equipment, electric, group
Facebook: information.ON, ABOUT.recent(events), computer
hype: fame, fun, idea, popular, surprise
numpty: bad, lazy, stupid, lack(work), dull

A weakness of our method is that in some cases it also adds noise in the generated features. For example, features such as *sleep* or *sing* generated for the verb *eat* are not ones we would expect to be part of the definition of *eat* (even if in a broader sense they might be related). Inappropriate features like this may be eliminated manually from the representations generated by the model. The model can thus also be used as an aid in a semi-automatic semantic resource creation/extension process proposing an initial representation that can be cleaned manually for applications that require a high-precision lexical semantic representation. Otherwise, the generated semantic features can be used in models performing some downstream task even without filtering out the noise. In that case, the added semantic features may improve the performance of the downstream tool providing mostly useful features for words that otherwise would completely lack semantic representation.

Quantitative analysis

We also carried out two kinds of quantitative analysis of the performance of our model. First, we checked the robustness of the model by performing a sanity check. For each word present in the original 4lang dictionary, we calculated how many of the semantic features present in the original definition were retrieved among the top N features returned by the model (feature recall, R_f) and the percentage of words for which all features were retrieved (word recall, R_w). The results are shown in Table 4 as a function of N (numbers are percentages). Recall was also calculated ignoring words having more than N features (R^p_w) and discounting features over the N limit for words having more than N features (R^p_f). As no definition contained more than 10 terms, R^p_w is identical to R_w and R^p_f is identical to R_f for $N \geq 10$. The definitions are terse and contain a minimal description for each word: for half of the words containing only a single term, and for almost all words not more than 5, see column $|f| \leq N$). Feature precision ($P(f)$) apparently decreases quickly as the number of features retrieved increases if we blindly accept only terms present in the original definitions as correct. See, however, further discussion below. The last column of the table shows the mean average precision (MAP) of features (terms) present in the original definitions.

N	R_w	R^p_w	R_f	R^p_f	$\|f\| \leq N$	$P(f)$	**MAP**
1	44.11	88.18	50.79	92.66	50.02	92.66	92.66
5	86.88	87.75	91.38	92.26	99.00	56.70	89.66
10	93.39	93.39	95.97	95.97	100.00	32.70	90.56
15	95.61	95.61	97.36	97.36	100.00	22.89	90.77
20	96.48	96.48	97.93	97.93	100.00	17.54	90.82

Table 4: Performance of the model for English tested on definitions in the 4lang vocabulary as a function of the number N of top-ranked features retrieved for each word. R_w: Word recall (words for which all features were retrieved), $R_w(poss)$: recall for words having no more than N features, R_f: feature recall, $R_f(poss)$: feature recall ignoring features over the top N, $|f| \leq N$: percentage of words having no more than N features, $P(f)$: feature precision, MAP: mean average precision of features. Numbers are percentages.

Language	**acc**	**d-acc**	**#F**	**#B**
English	75.13%	90.07%	559	277
Hungarian	73.86%	88.34%	584	295

Table 5: Performance of the model on 280 different test words for English and Hungarian. **acc**: feature accuracy, **d-acc**: domain accuracy of features, **#F**: different features, **#B**: features marked wrong at least once.

In the other experiment, we selected 280 words not present in the original dictionary randomly from a predefined list of Hungarian words in which each word was assigned to one of 28 semantic domains (e.g. food, vehicles, locations, occupations, etc.). From each domain 10 words were chosen randomly and were translated to English. Then, for these words, the 10 nearest features were generated and two human annotators checked whether each feature was adequate for each given word. The same evaluation was performed for Hungarian. The agreement ratio between the annotators was 0.798 for English and 0.734 for Hungarian according to Cohen's kappa, which is substantial in both cases. The results are shown in Table 5.

The table shows feature accuracy (acc: the ratio of correctly assigned features) in each domain. We also automatically computed feature "domain accuracy" (d-acc): here we ignored feature assignment errors where the same feature was marked adequate for another test word in the same domain. The number of different features that appeared in

this evaluation and the number of features marked wrong at least once are shown in the last two columns. Note that the feature accuracy (precision) for 10 features retrieved turned out to be much higher (75.13%) than in the sanity check experiment (only 32.70%) even though this list contained words not in the original resource. The reason for this is that the model returns many features which, while not explicitly present in the original terse definitions, correctly follow from the knowledge embodied in the feature model. E.g. While the definition of *dog* in 4lang contains only 3 terms: *animal, faithful* and *carnivorous*, the top 10 features retrieved from the model also include *mammal, HAS{four(legs)}, hairy* and *companion*. The sanity check experiment thus grossly underestimated the precision of the model.

Conclusion

We have presented an automatic method that is able to assign semantic features to words of natural language. This approach exploits the representative power of neural word embeddings by mapping features derived from formal definitions of words to the vector space of the given language. In addition to some illustrative examples, we have presented the evaluation of the models demonstrating that the method works with relatively high accuracy. Although there is a moderate amount of noise in the set of generated features, the method has a very high coverage, being able to process proper names or non-standard words as well, which cannot all be included in hand-made static knowledge bases. As such, our automatic method can be used as the base of a manually constructed resource, or can provide valuable input for downstream applications, such as commonsense inference systems.

Acknowledgments

This research has been implemented with support provided by grant FK125217 of the National Research, Development and Innovation Office of Hungary financed under the FK17 funding scheme.

References

Baroni, M.; Dinu, G.; and Kruszewski, G. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 238–247. Baltimore, Maryland: Association for Computational Linguistics.

Brown, S. W. 2008. Choosing sense distinctions for wsd: Psycholinguistic evidence. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, 249–252. Stroudsburg, PA, USA: Association for Computational Linguistics.

Davis, E. 1990. *Representations of commonsense knowledge*. Morgan Kaufmann.

Fellbaum, C., ed. 1998. *WordNet: an electronic lexical database*. MIT Press.

Kornai, A.; Ács, J.; Makrai, M.; Nemeskey, D. M.; Pajkossy, K.; and Recski, G. 2015. Competence in lexical semantics. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, 165–175. Denver, Colorado: Association for Computational Linguistics.

Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; and Bizer, C. 2015. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal* 6(2):167–195.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, 3111–3119.

Mikolov, T.; Yih, W.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, 746–751.

Miller, G. A. 1995. Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM* 38:39–41.

Novák, A.; Siklósi, B.; and Oravecz, C. 2016. A New Integrated Open-source Morphological Analyzer for Hungarian. In Chair), N. C. C.; Choukri, K.; Declerck, T.; Goggi, S.; Grobelnik, M.; Maegaard, B.; Mariani, J.; Mazo, H.; Moreno, A.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris, France: European Language Resources Association (ELRA).

Novák, A. 2014. A new form of humor – mapping constraint-based computational morphologies to a finite-state representation. In Chair), N. C. C.; Choukri, K.; Declerck, T.; Loftsson, H.; Maegaard, B.; Mariani, J.; Moreno, A.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA).

Orosz, G., and Novák, A. 2013. PurePos 2.0: a hybrid tool for morphological disambiguation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013)*, 539–545. Hissar, Bulgaria: INCOMA Ltd. Shoumen, BULGARIA.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Pereira, F.; Tishby, N.; and Lee, L. 1993. Distributional clustering of english words. In *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, ACL '93, 183–190. Stroudsburg, PA, USA: Association for Computational Linguistics.

Siklósi, B. 2016. Using embedding models for lexical categorization in morphologically rich languages. In Gelbukh, A., ed., *Computational Linguistics and Intelligent Text Processing: 17th International Conference, CICLing 2016*. Konya, Turkey: Springer International Publishing, Cham.

Sokal, R. R., and Rohlf, F. J. 1962. The comparison of dendrograms by objective methods. *Taxon* 11(2):33–40.

Summers, D. 2005. *Longman Dictionary of Contemporary English*. Longman Dictionary of Contemporary English Series. Longman.

Toutanova, K.; Klein, D.; Manning, C. D.; and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, 173–180. Stroudsburg, PA, USA: Association for Computational Linguistics.

van der Maaten, L., and Hinton, G. E. 2008. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research* 9:2579–2605.

Ward, J. H. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association* 58(301):236–244.