Assessing the impact of Word Embeddings for Relation Classification: An Empirical Study

Dzhal Antonov[1,3,†], Davide Buscaldi[3,*,†]

[1]Ecole Polytechnique, 91120 Palaiseau, France
[2]Higher School of Economics, Moscow, Russia
[3]LIPN, Université Sorbonne Paris Nord, CNRS UMR 7030, 93430 Villetaneuse, France

Abstract

This paper presents an empirical study on the value of concept embeddings for relation classification, with a particular focus on hypernymy detection. Knowledge Graphs have become popular due to their ability to represent knowledge in a standardized form that enables reasoning, but often suffer from incompleteness. Relation Classification is a task that can help alleviate this issue. Recent advances in Deep Learning applied to Natural Language Processing have provided researchers with powerful tools for detecting relations between concepts: word embeddings. We investigate the effectiveness of different types and combinations of embeddings for the automatic relation classification task. We conduct experiments on two datasets based on WordNet and the AI-KG Knowledge Base. Our results confirm previous results that it is challenging to deduce the semantic relations from embeddings alone. We observe that hypernymy cannot be captured solely by a sub-space of the embedding space, despite specific dimensions carrying more information about this relation than others. Additionally, we show that it is difficult to apply a model learned on a general ontology to other domains, and that imbalance problems are aggravated in large knowledge bases where one relation dominates over all the others.

Keywords

Knowledge Graphs, Word Embeddings, Relation Classification

1. Introduction

Knowledge Graphs (KGs) have become very popular in various tasks, thanks to their versatility and their ability to represent knowledge in a standardized form that enables reasoning, both in open and closed-domain applications. Examples of public, cross-domain knowledge graphs that encode common knowledge include DBpedia [1], and YAGO [2]. However, KGs often suffer from incompleteness problems [3]. Relation Classification is a task that can help alleviate the problem of incompleteness in KGs; it consists in determining the presence of a named relation between two semantic entities.

TEXT2KG 2023: Second International Workshop on Knowledge Graph Generation from Text, May 28 - Jun 1, 2023, co-located with Extended Semantic Web Conference (ESWC), Hersonissos, Greece

[*] Corresponding author.

[†] These authors contributed equally.

dvantonov@edu.hse.ru (D. Antonov); buscaldi@lipn.fr (D. Buscaldi)
https://github.com/chejuro/ (D. Antonov); https://lipn.fr/~buscaldi (D. Buscaldi)
0000-0003-1112-3789 (D. Buscaldi)

CEUR Workshop Proceedings (CEUR-WS.org)

Recently, the outstanding achievements of Deep Learning applied to Natural Language Processing provided researchers with powerful tools for the detection of relations between concepts. In particular, the significance of word embeddings in present-day Natural Language Processing (NLP) techniques cannot be overstated. Besides their usefulness in encoding textual data in neural network models, they are crucial due to their ability to capture a substantial amount of linguistic and semantic information. Word2Vec [4] showed that it is possible to encode some kind of relational information within the embedding space, in particular the ability to capture word analogies (e.g. $king - man + woman \simeq queen$). However, some works such as [5, 6] proved that the expectations regarding relation prediction from the entity embeddings were excessively optimistic. Approaches such as TransE [7] have somehow “fixed" the problem by introducing techniques to modify the position of concepts in the embedding space depending on the relations in which they occur in the Knowledge Base. However, such methods can still consider invalid relations that they have not seen in the knowledge base used for training, despite them being correct.

Some works have shown that it is actually possible to discover relations by looking at the embedding of the concepts. Kata Gabor et al. [8] tried various combinations of the entities concept to see which ones were the most useful to predict relations. They proved that the vector offset method for analogies is the least efficient in capturing generic semantic relations at a large scale, while pairwise similarities can be better exploited in an additive or concatenative setting. Maurizio Atzori and Simone Balloccu proposed an algorithm to deduce the existence of the hypernymy (or *is_a*) relation between two concepts from their word embeddings [9]. Their work tries to unentangle the contextual information contained in the embedding space. They obtained the best accuracy in hypernym discovery for unsupervised systems on data from SemEval-2018 [10].

In this work, we carry out an empirical study, in the wake of these works, trying to understand what is the value of concept embeddings for relation classification. Despite the diffusion and the interest of this task, only a few works have tried to carry out an experimental study on the role of embeddings for relation classification, and always on specific types of relations: for instance, temporal [11] and discourse-based relations [12]. We considered two datasets: a general one and a domain-specific one, both including the hypernymy relation. This consideration was important as something we wanted to verify is if it is possible to learn a hypernymy detection model from a general dataset and apply it to a domain-specific one, and what would be the accuracy loss in doing so. In the rest of the paper, we present in Section 2 the dataset used, in 3 the different types of embeddings used and in 4 the experiments carried out. Finally, we present in 5 our conclusions from these experiments. The code used for this work is available at the address https://github.com/chejuro/Relation-prediction-from-embeddings.

2. Knowledge Bases and Datasets

2.1. WordNet and WN18RR

The WN18 dataset was introduced in 2013 by Bordes et al. [13]. It included the full 18 relations scraped from WordNet for roughly 41,000 synsets (the equivalent of a concept in WordNet). This dataset was affected by leakage due to the presence of symmetric relations, therefore in 2018 [14] introduced the WN18RR dataset. This dataset features 11 relations, no pair of which is reciprocal. Wordnet is a manually-curated resource of semantic concepts, restricted to more "linguistic" relations compared to those expected in general world knowledge. The most common semantic relation is the hypernymy one, representing more than 45% of total relations, while some relations are very rare, such as antonymy (1.2%) or attribute (0.4%) [15]. The number of synsets is more than $175,000$ in the latest version.

2.2. Artificial Intelligence Knowledge Graph

AI-KG [16] is a large-scale automatically generated knowledge graph that is made up of a large number of articles and describes around 2.3M triplets that are connected by 55 semantic relations. These relations include hypernymy (encoded as the *skos:broader* relation) and others related to the scientific discourse (for instance: *method A* uses *resource B* to perform *task C*. Similarly to Wordnet, AI-KG also has some reciprocal relations and an imbalance problem. After cleaning and filtering out the least frequent relations we keep 10 relations with $1M$ triplets. The details about the relations are shown in Table 1.

Table 1
Relations distribution in the AI-KG dataset

Relation	Triples
methodUsedBy	460724
otherEntityUsedBy	136310
OtherEntityIncludedBy	113678
hypernym	107812
materialUsedBy	41075
methodIncludedBy	30332
OtherEntityPredictedBy	29382
taskUsedBy	22341
methodEvaluatedBy	18622
materialIncludedBy	11295

3. Word Embeddings

3.1. Word2Vec

Word2Vec [4] is one of the most famous and widely spread word vectorization techniques. Word2Vec embeddings are learned either using the CBOW (Continuous Bag-of-Words) or skip-gram model. In the CBOW approach, the model predicts the target word given a context of surrounding words. The context words are summed together to form a continuous bag-of-words representation, which is then used as input to the model to predict the target word. The CBOW algorithm is trained by iteratively adjusting the word embeddings to improve the accuracy of the predicted target word given its context. Compared to the skip-gram model, which predicts the context words given a target word, CBOW is faster to train.

3.2. GloVe (Global Vectors)

GloVe [17] is another model of non-deep representation of words, proposed in 2014 by a group of developers at Stanford. It uses a co-occurrence matrix that describes how frequently different words appear together in a corpus of text. The co-occurrence matrix is then factorized to obtain word embeddings that capture the semantic and syntactic relationships between words. GloVe is able to capture both local and global relationships between words, and it has been shown to perform well on a wide range of NLP tasks, including language modeling, sentiment analysis, and machine translation.

3.3. FastText

FastText is an extension to Word2Vec proposed by Facebook in 2016. It addresses an important problem with embeddings, the Out-Of-Vocabulary (OOV) words. With GloVe and Word2Vec, if a word is not known, then it is not possible to obtain an embedding. FastText addresses the problem by breaking words into several character n-grams (sub-word tokens). Therefore, it is possible that even OOV words can be reconstructed by assembling sub-word tokens.

3.4. ConceptNet Numberbatch

ConceptNet Numberbatch [18] are another type of Word2Vec-like embeddings trained on a heterogeneous set of sources, including Wikipedia and ontologies such as OpenCyc and even WordNet. The concept repository is based on ConceptNet, a semantic network that encodes general knowledge about the world in a machine-readable format. These embeddings have been conceived with the objective to include both structured and unstructured data sources, which allows them to capture both explicit and implicit relationships between words. Therefore, these embeddings address the flaws highlighted by [5] and [6] regarding the ability of word embeddings to capture semantic relations between words.

3.5. BERT

BERT [19] is a bidirectional transformer-based machine learning technique pretrained using a combination of masked language modeling objective and next sentence prediction on a very large corpus. The BERT model is based on the Transformer model [20], which includes the attention mechanism that highlights the contextual relationship between the words in a phrase. The basic part consists of an encoder to read the input text and a decoder to generate a prediction, filling the masked parts of the training sentence. In order for BERT to create a language representation model, it only needs the encoder part. The encoder input to BERT is a sequence of tokens that are converted into vectors and then processed in a neural network. The main advantage of BERT embeddings is that they are contextual, i.e. a word does not have a fixed embedding but it changes in the function of its context. BERT embeddings can also be computed for OOV words as they use sub-word tokenization, similarly to FastText.

3.6. Sentence-BERT

One of the problems of word embeddings is that they are representing a single word or a compound word (if this compound word has been labeled as such in the training corpora). If a concept is represented by multiple words, the usual way to obtain a representation is by averaging or maxing over the embedding dimensions. But this method has been proven to be sub-optimal. Sentence-BERT [21] is a modified version of the pre-trained BERT network that creates comparably meaningful sentence embeddings utilizing a cosine similarity or triplet loss on top of a siamese network architecture. To create a fixed-size sentence embedding, Sentence-BERT adds pooling to the token embeddings produced by BERT. This network is capable of encoding phrase semantics and therefore it can be used to encode concepts that are expressed using multiple words.

4. Classification experiments

Given a pair of entities or concepts (h, t) extracted from a Knowledge Base K, our objective is to predict a relation r such as (h, r, t) is a valid relation in K. Therefore, we can treat the problem as a multi-class classification task in which, given the representation for the (h, t) pair, we predict the target r. We select from the dataset a training set (80$\%$ of triplets) in which r is known for each pair of concepts and use the rest of the dataset as a validation set.

There are various possibilities regarding how to combine the representations (i.e. the embeddings) of the h and t together. According to the study by [8], analogy combinations do not work well; therefore we considered averaging and concatenation. Given $emb(h) = (x_1, \ldots, x_n)$ and $emb(t) = (y_1, \ldots, y_n)$, with n embedding size, for averaging we obtain $emb(h, t)$ as the pairwise mean: $emb(h, t) = \left(\frac{x_1+y_1}{2}, \frac{x_2+y_2}{2}, \ldots, \frac{x_n+y_n}{2}\right)$. For concatenation, $emb(h, t) = (x_1, \ldots, x_n, y_1, \ldots y_{2n})$. The n size of the embeddings varies depending on the model. For word2vec and derived models it is 300 dimensions, while for BERT and S-BERT is 768.

4.1. WN18RR dataset

4.1.1. Logistic Regression model

Logistic regression was chosen as a basic classification algorithm because it has a lot of advantages in comparison with other algorithms: it is efficient to train, effective for multi-class problems, and it can estimate feature importance by model coefficients. Table 2 and Table 3 present results of classification for every word embeddings approach with pairwise mean operation and with vector concatenation. Accuracy is calculated as the correct prediction over the total number of examples, independently from the class. Precision is calculated as the macro-average precision (TP/(TP+FP)) over each class. F1-score is the harmonic mean between macro-average precision and macro-average recall (TP/(TP+FN)). According to the results, the concatenation operation works significantly better and this can be explained by the fact that we save the information of each entity in the final vector, and with the mean operation, we mix that information. This result confirms the conclusions by [8].

Table 2
Results for the Logistic Regression model with vector pairwise mean operation. WN18RR dataset.

Embedding type	Accuracy	Precision (macro)	F1-score (macro)
Word2Vec	0.7920	0.6549	0.60
FastText	0.7970	0.6937	0.5988
GloVe	0.7910	0.7008	0.6038
ConceptNet Numberbatch	0.7945	0.6865	0.6078
BERT	0.8348	0.7312	0.6493
Sentence-BERT	0.8481	0.7377	0.6589

Table 3
Results for the Logistic Regression model with vector concatenation. WN18RR dataset.

Embedding type	Accuracy	Precision (macro)	F1-score (macro)
Word2Vec	0.8629	0.7876	0.7170
FastText	0.8686	0.7817	0.7044
GloVe	0.8516	0.7478	0.7003
ConceptNet Numberbatch	0.8717	0.7769	0.7224
BERT	0.9013	0.8020	0.7504
Sentence-BERT	0.9297	0.7984	0.7477

4.1.2. Logistic Regression interpretation for hypernymy relation

We considered the concatenation representation to derive an interpretation of the embeddings from the Logistic Regression model and we performed feature selection. This selection can highlight the most important features, that is dimensions in the embedding space that are expected to carry the most information regarding the hypernymy relation. Thus we trained a new logistic regression model on a truncated set of features and compared the results with those obtained with the model based on all features.

We extracted the ten most important features (see Table 4) from the logistic regression model based on ConceptNet-NB embeddings. It is interesting to observe that only 2 of these features are related to the subject of the relation (the hyponym or more specific word) while the others are related to the object (the hypernym). It is difficult to tell what the individual features mean as each dimension of the embeddings is not linked to any specific linguistic feature. We used these 10 best features to retrain a new “compressed" logistic regression model. We obtained a precision of 0.7227 on the validation set while with all 600 features, it was 0.83426. According to this result, the hypernymy relation is not encoded by a subset of dimensions (although some are more informative than others), but rather the full embedding is required to capture the meaning of the semantic relation.

4.1.3. Multi-Layer Perceptron (MLP)

Next, we implemented Multi-Layer Perceptron as a deep model baseline. It consists of 5 to 10 fully connected linear layers, with alternating hidden layers of size 300 and 100 and a final layer of N units, corresponding to the number of relations to predict. The loss is cross-entropy loss. Between each layer we have an activation function and a dropout layer (dropout=0.2). Table 5

Table 4
Study on hypernym feature importance (logistic regression coefficients). Each feature id correspond to the position in the embedding. Features with $0 \leq id_f < 300$ are from the embedding of the subject word (the hyponym or more specific word). Features with $id_f \geq 300$ are from the embedding of the object word (the hypernym).

feature_id	weight	source emb.
315	3.269	object
423	2.619	object
508	2.570	object
354	2.491	object
255	2.466	subject
538	2.345	object
404	2.269	object
566	2.241	object
280	2.191	subject
379	2.166	object

shows the different hyperparameters of the model and the number of hidden layers. ReLu is also tested, but the outcome does not change too much, even if it is the best one. According to the results, MLP works much better than logistic regression.

Table 5
Study on hyperparameters effect of MLP model over WN18 data and NumberBatch embeddings.

depth	activation	dropout	learning rate	epoch	precision
5	sigmoid	0.2	0.001	10	0.9216
5	Relu	0.2	0.001	10	0.9255
8	sigmoid	0.2	0.001	10	0.9188
10	sigmoid	0.2	0.001	10	0.9204

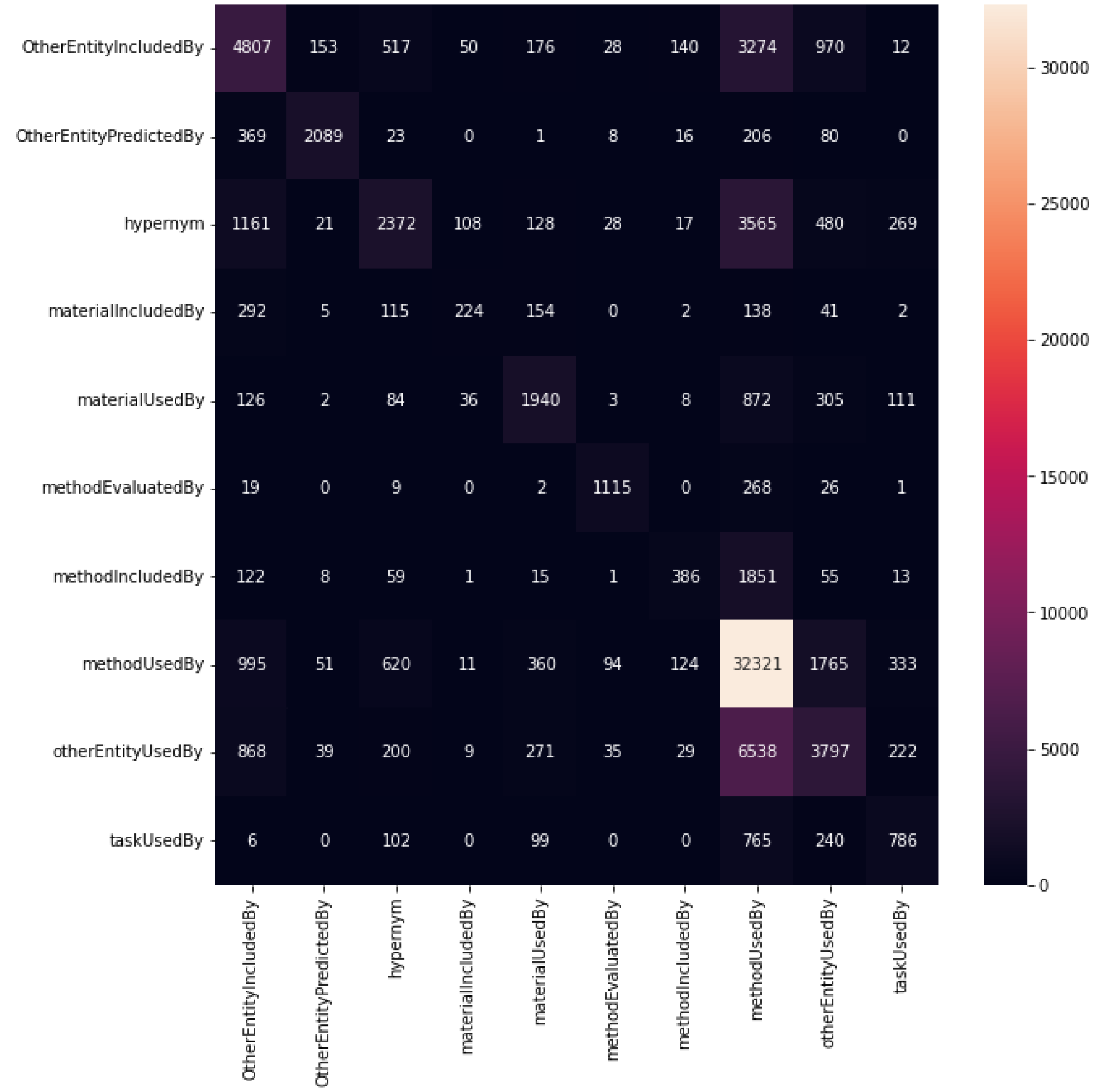
4.2. AI-KG dataset

4.2.1. Classification models

The AI-KG dataset with 10 relations and $971,571$ triplets was divided into train and validation sets of size 90% and 10% respectively. Similarly to the WordNet experiments, we trained Logistic Regression and MLP models and they obtained 0.5365 and 0.6197 precision scores respectively. In Figure 1 we show the confusion matrix for the MLP model.

4.2.2. Class imbalance problem

We can notice from the confusion matrix that there is a class imbalance problem. As we saw in Table 1, there is a large difference in the number of triplets supported by different relations. To overcome this problem we tried to group together some relations to reduce the problem to 5 relations. We did this as in AI-KG some relations are semantically similar while their difference is only the category of one of the entities involved. For instance: methodUsedBy,



	OtherEntityIncludedBy	OtherEntityPredictedBy	hypernym	materialIncludedBy	materialUsedBy	methodEvaluatedBy	methodIncludedBy	methodUsedBy	otherEntityUsedBy	taskUsedBy
OtherEntityIncludedBy	4807	153	517	50	176	28	140	3274	970	12
OtherEntityPredictedBy	369	2089	23	0	1	8	16	206	80	0
hypernym	1161	21	2372	108	128	28	17	3565	480	269
materialIncludedBy	292	5	115	224	154	0	2	138	41	2
materialUsedBy	126	2	84	36	1940	3	8	872	305	111
methodEvaluatedBy	19	0	9	0	2	1115	0	268	26	1
methodIncludedBy	122	8	59	1	15	1	386	1851	55	13
methodUsedBy	995	51	620	11	360	94	124	32321	1765	333
otherEntityUsedBy	868	39	200	9	271	35	29	6538	3797	222
taskUsedBy	6	0	102	0	99	0	0	765	240	786

Figure 1: Multi-Layer perceptron confusion matrix on AI-KG dataset in the 10 relations setup.

otherEntityUsedBy, materialUsedBy, taskUsedBy were grouped in just one class *usedBy*. In this way, we got a more balanced situation with 5 classes and consequently got slightly better results: 0.7811 in precision. The confusion matrix for this reduced dataset is shown in Figure 2. We also show in Table 6 the precision and recall scores for each relation.

4.2.3. Hypernymy relation prediction

Another hypothesis that we wanted to test was to train a model on WordNet (general knowledge) and see if this model could be applied to the more domain-specific data (AI-KG) to predict the

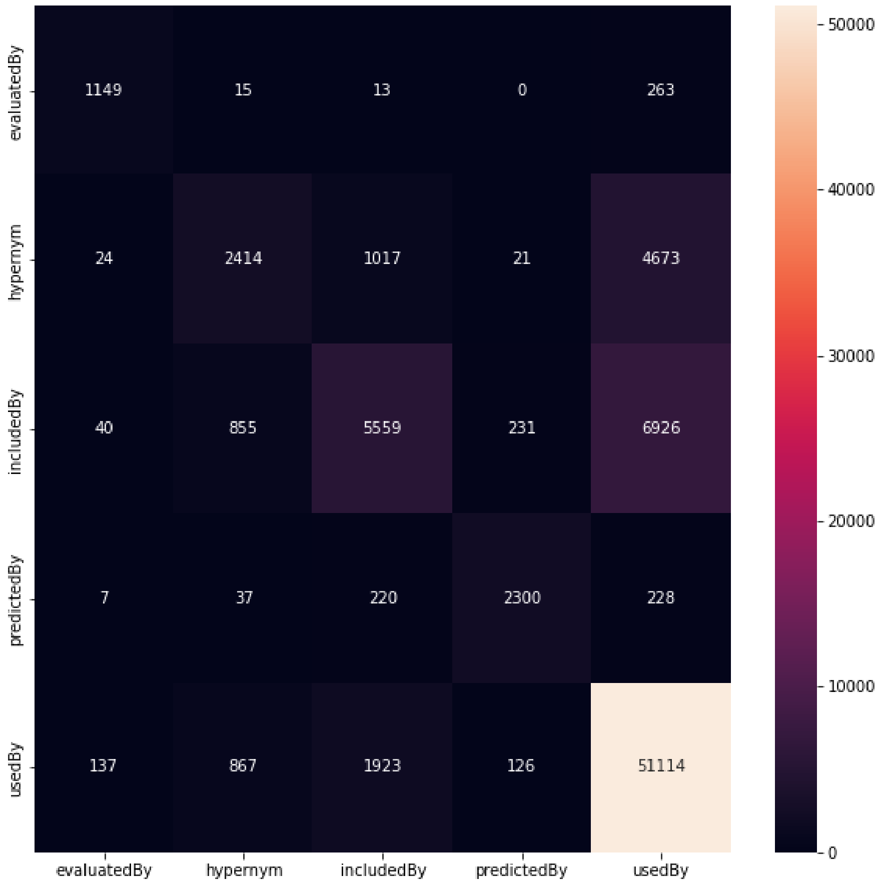


Figure 2: Multi-Layer perceptron confusion matrix on AI-KG dataset in the 5 relations setup.

Table 6
5-relations classification results using the MLP model with ConceptNet-NumberBatch embeddings.

relation	precision	recall
evaluatedBy	0.798	0.847
hypernym	0.296	0.576
includedBy	0.408	0.637
predictedBy	0.823	0.859
usedBy	0.944	0.801

existence of a hypernymy relationship between two entities. The hypernymy relationship can be mapped to the “skos:broader" relation in AI-KG. In Figure 3 we can see that similar concepts are arranged in a similar way in the two knowledge graphs, which would suggest a certain compatibility of the relations learned in WordNet with those present in AI-KG. However, the result of the best model obtained only 0.13 in precision. It is still better than random choice because in the AI-KG dataset there are 80425 hypernymy relations and 721,156 non-hypernymy relations which means that random choice to predict correctly is around 11%. However, this

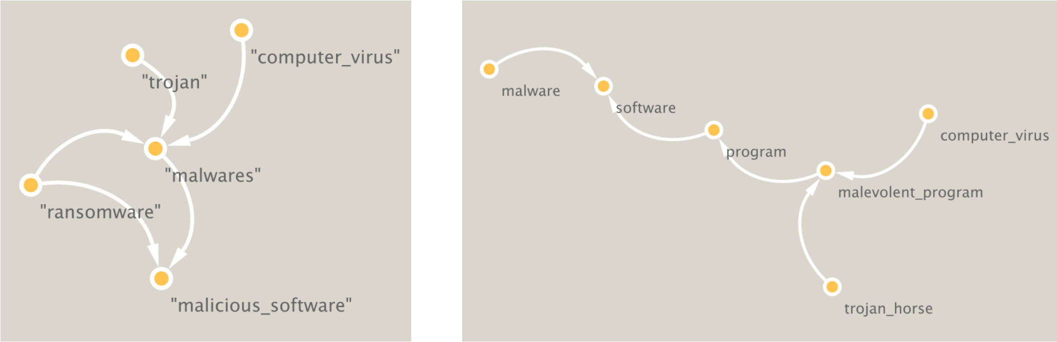


Figure 3: Excerpts of taxonomies regarding computer viruses, malware and trojans in AI-KG (left) and WordNet (right).

is an interesting result that lays some doubts about the generalization capability of relation prediction models.

5. Conclusions

In this work, we tested various embeddings types and combination of embeddings for the automatic relation classification task. The classification has been tested on two datasets based on the WordNet and AI-KG Knowledge Bases. Our results confirm what emerged in previous works that is difficult to extrapolate semantic relations from embeddings alone. As our experiment with reduced dimensions shows, hypernymy cannot be captured just by a sub-space of the embedding space, even if some dimensions seem to carry more information regarding this relation than other ones. We also showed how imbalance problems may affect relation classification in some more specific knowledge bases such as AI-KG, and that models built to predict a relation on a general knowledge base cannot be used to predict the same relation on a different more specific knowledge base. We plan in future to extend this experimentation to further embeddings and Knowledge Bases.

Acknowledgments

This project has been carried out as part of the “projet d’établissement" titled “Plongements lexicaux pour l’acquisition et la structuration de connaissances", funded by Sorbonne Paris Nord University.

References

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in: K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, P. Cudré-Mauroux (Eds.), The Semantic Web, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 722–735.

[2] F. M. Suchanek, G. Kasneci, G. Weikum, Yago: A large ontology from wikipedia and wordnet, Journal of Web Semantics 6 (2008) 203–217.

[3] M. Destandau, J.-D. Fekete, The missing path: Analysing incompleteness in knowledge graphs, Information Visualization 20 (2021) 66–82. URL: https://doi.org/10.1177/1473871621991539. doi:10.1177/1473871621991539. arXiv:https://doi.org/10.1177/1473871621991539.

[4] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, Advances in Neural Information Processing Systems 26 (2013).

[5] T. Linzen, Issues in evaluating semantic spaces using word analogies, in: Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 13–18. URL: https://aclanthology.org/W16-2503. doi:10.18653/v1/W16-2503.

[6] A. Rogers, A. Drozd, B. Li, The (too many) problems of analogical reasoning with word vectors, in: Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 135–148. URL: https://aclanthology.org/S17-1017. doi:10.18653/v1/S17-1017.

[7] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in neural information processing systems 26 (2013).

[8] K. Gábor, H. Zargayouna, I. Tellier, D. Buscaldi, T. Charnois, Exploring vector spaces for semantic relations, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 1814–1823. URL: https://aclanthology.org/D17-1193. doi:10.18653/v1/D17-1193.

[9] M. Atzori, S. Balloccu, Fully-unsupervised embeddings-based hypernym discovery, Information 11 (2020) 268. doi:10.3390/info11050268.

[10] J. Camacho-Collados, C. Delli Bovi, L. Espinosa-Anke, S. Oramas, T. Pasini, E. Santus, V. Shwartz, R. Navigli, H. Saggion, SemEval-2018 task 9: Hypernym discovery, in: Proceedings of the 12th International Workshop on Semantic Evaluation, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 712–724. URL: https://aclanthology.org/S18-1115. doi:10.18653/v1/S18-1115.

[11] P. Mirza, S. Tonelli, On the contribution of word embeddings to temporal relation classification, in: The 26th International Conference on Computational Linguistics, ACL, 2016, pp. 2818–2828.

[12] C. Braud, P. Denis, Comparing word representations for implicit discourse relation classification, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 2201–2211.

[13] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data 2013 (2013).

[14] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, CoRR abs/1707.01476 (2017). URL: http://arxiv.org/abs/1707.01476. arXiv:1707.01476.

[15] M. Maziarz, M. Piasecki, S. Szpakowicz, The chicken-and-egg problem in wordnet design:

Synonymy, synsets and constitutive relations, Lang. Resour. Eval. 47 (2013) 769–796. URL: https://doi.org/10.1007/s10579-012-9209-9. doi:10.1007/s10579-012-9209-9.

[16] D. Dessì, F. Osborne, D. Reforgiato Recupero, D. Buscaldi, E. Motta, H. Sack, AI-KG: an automatically generated knowledge graph of artificial intelligence, in: The Semantic Web–ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part II 19, Springer, 2020, pp. 127–143.

[17] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[18] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, CoRR abs/1612.03975 (2016). URL: http://arxiv.org/abs/1612.03975. arXiv:1612.03975.

[19] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, CoRR abs/1706.03762 (2017). URL: http://arxiv.org/abs/1706.03762. arXiv:1706.03762.

[21] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, CoRR abs/1908.10084 (2019). URL: http://arxiv.org/abs/1908.10084. arXiv:1908.10084.