ProsperAMnet at the FinSim Task: Detecting hypernyms of financial concepts via measuring the information stored in sparse word representations

Gábor Berend[1,2] , **Nobert Kis-Szabó**[1] , **Zsolt Szántó**[1]

[1]University of Szeged, Institute of Informatics
[2]MTA-SZTE Research Group on Artificial Intelligence
{berendg,ksznorbi,szantozs}@inf.u-szeged.hu

Abstract

In this paper we propose and carefully evaluate the application of an information theoretic approach for the detection of hypernyms for financial concepts. Our algorithm is based on the application of sparse word embeddings, meaning that – unlike in the case of traditional word embeddings – most of the coefficients in the embeddings are exactly zero. We apply an approach that quantify the extent to which the individual dimensions for such word representations convey the property that some word is the hyponym of a certain top-level concept according to an external ontology. Our experimental results demonstrate that substantial improvements can be gained by our approach compared to the direct utilization of the traditional dense word embeddings. Our team ranked second and fourth according to average rank score and mean accuracy that were the two evaluation criteria applied at the shared task.

1 Introduction

We introduce our contribution to the FinSim 2020 shared task [Maarouf *et al.*, 2020] where the task was to classify financial terms according to their ontological properties.

As sparse word embeddings have been reported to convey increased interpretability [Murphy *et al.*, 2012; Faruqui *et al.*, 2015; Subramanian *et al.*, 2018], we investigated the extent to which applying them can improve the extraction of financial taxonomic relations. To this end we carefully evaluate in this paper an algorithm in the task of extracting taxonomic relations for financial terms on the shared task dataset by exploiting an algorithm for extracting commonsense knowledge from sparse word representations proposed in [Balogh *et al.*, 2020]. Our results corroborate previous claims that the application of sparse word representations not only result in a more interpretable representation, but the systems built on top of them often outperform approaches that employ dense word embeddings [Faruqui *et al.*, 2015; Berend, 2017]. We release our source code and trained embeddings in order to foster reproducibility of our results[1].

2 Related work

Hypernym discovery has spurred substantial research attention with one of the 2018 SemEval shared task being focused on the detection of hypernyms in multiple languages and domains [Camacho-Collados *et al.*, 2018]. The top-performing system applied a combination of supervised learning and unsupervised pattern matching techniques [Bernier-Colborne and Barrière, 2018]. [Held and Habash, 2019] also argued for the applications of hybrid approaches involving Hearst patters [Hearst, 1992] for extracting hypernyms. Most recently, [Dash *et al.*, 2020] introduced Strict Partial Order Networks (SPON), a neural network architecture for detecting word pairs for which the `IsA` relation holds paying special attention to the fact of the relation being asymmetric.

[Berend *et al.*, 2018] employed sparse word representations and formal concept analysis for building a model that decides if a word is a hypernym of another by investigating the non-zero coefficients for a pair of input expressions. Even though our work also exploits sparse word representations, we rather build our framework on an information theory-inspired approach that we introduce in the followings.

3 System description

Our framework adapts recent algorithm in [Balogh *et al.*, 2020] which devises an information theory-inspired algorithm for quantifying the extent to which the individual dimensions of sparse word representations relate to certain commonsense properties of concepts. The basis of the algorithm is to measure the amount of information overlap between the properties of concepts and the nonzero coefficients of sparse word representations. [Balogh *et al.*, 2020] took ConceptNet [Speer and Havasi, 2012] as the basis for measuring the information overlap, however, the approach is generalizable to any commonsense knowledge.

We next summarize our approach in details. As a first step, we extract the raw text from the prospectuses that were provided by the organizers in pdf format using Tika. As a subsequent step, we trained standard static word embeddings using approaches fasttext [Bojanowski *et al.*, 2017] and Glove [Pennington *et al.*, 2014].

We relied on the default tokenization protocol and set all the hyperparameters of the algorithms for creating the embeddings to their default settings as well in order to avoid ex-

[1] `https://github.com/begab/prosperAM-finsim`

cessive hyperparamter tuning. In the end, we were left with a vocabulary of 17,105 unique word forms and 100 dimensional dense embeddings.

Our next step was to derive the sparse word representations from the dense embeddings that we created earlier. For this step, we relied on the algorithm proposed in [Berend, 2017], that is given matrix $X \in \mathbb{R}^{n \times m}$ $(n = 17,105, m = 100)$ containing a collection of stacked dense embeddings of cardinality n, we strive to solve

$$\min_{\alpha \in \mathbb{R}^{n \times k}_{\geq 0}, D \in \mathbb{R}^{k \times m}} \frac{1}{2}\|X - \alpha D\|_F^2 + \lambda\|\alpha\|_1, \qquad (1)$$

with the additional constraint that the vectors comprising D have a bounded norm. That is, we would like to express each dense word embedding as a sparse linear combination of the vectors included in D. The number of vectors included in $D \in \mathbb{R}^{k \times m}$ is controlled by the value of k. We conducted experiments for $k \in \{1000, 1500, 2000\}$.

The ℓ_1-based penalty term included in (1) causes most of the coefficients in α to be zero, and we tread the rows of this matrix as our sparse word representations. Larger values for the regularization coefficient λ results in higher levels of sparsity in the word representation that we obtain. We performed our experiments with $\lambda \in \{0.1, , 0.2, 0.3, 0.4, 0.5\}$. For solving (1), we used the dictionary learning algorithm introduced in [Mairal *et al.*, 2009].

Next, we constructed the matrix of representations for the terms in the training dataset. This resulted in a matrix of $T \in \mathbb{R}^{100 \times k}$, with 100 referring to the number of terms included in the training dataset. The embeddings for multi-token terms got determined by taking the centroid of the vectorial representation of the words that are included in a multi-token expression.

We subsequently constructed a binary matrix $B \in \{0, 1\}^{8 \times 100}$. In this matrix, every row corresponds to one of the eight labels, i.e., {Bonds, Forward, Funds, Future, MMIs, Option, Stocks, Swap} and an entry b_{ij} was set to 1 if training term j was labeled by label i in the training data and 0 otherwise.

By multiplying matrices B and T we obtained such a matrix $M \in \mathbb{R}^{8 \times k}$ which includes the sparse coefficients of the terms aggregated by the labels they belong to. We treated this matrix as an incidence matrix and calculated the normalized positive pointwise mutual information [Bouma, 2009]. For some label l_i and dimension d_j, we calculated this quantity (that we abbreviate as NPPMI) as

$$\text{NPPMI}(l_i, d_j) = \max\left(0; \ln\frac{P(l_i, d_j)}{P(l_i)P(d_j)} \Big/ -\ln P(l_i, d_j)\right)$$

In the above formula $P(l_i)$ refers to the probability of observing label i, $P(d_j)$ indicates the probability of dimension j having a non-zero value and $P(l_i, d_j)$ refers to the joint probability of the two events. We derived these probabilities by taking the row and column marginals of the ℓ_1-normalized version of the incidence matrix M. By performing NPPMI over every entry of M, we obtain matrix $A \in [0, 1]^{8,k}$, every entry of which determines the strength of association between label i and dimension j.

When facing a new term that is associated by vector $\mathbf{v} \in \mathbb{R}^k$, we take the product $\mathbf{s} = A\mathbf{v}$. An element s_i from $\mathbf{s}$ can be regarded as a score indicating the extent to which $\mathbf{v}$ refers to a vector that describes a term that belong to label i. Our final prediction hence is going to be label i^* for which $i^* = \arg\max_i s_i$.

4 Experiments

We first report our experiments that we obtained for our official submissions in the shared task. During this batch of experiments, we were working with 100 dimensional fasstext vectors created based on the training data provided by the shared task organizers, using the CBOW training approach with the default hyperparamter settings. We used the training set as the development set by measuring the performance of our algorithm over the 100 training instances in a leave-one-out fashion, i.e. averaged the evaluation metrics on every training term, while excluding the currently evaluated term from building our model.

For evaluating purposes we used the two official measures for the shared task, i.e. Mean Accuracy (MA) and Average Rank (AR). MA quantifies the percentage of terms for which a model regarded the true class label as the most likely one, whereas AR also takes into consideration the position of the correct label within the ranked list of class labels for an individual term. For the MA metric higher values mean better performance, whereas AR behaves in the opposite manner.

4.1 Centroid-based baseline

In order to see the added value of using sparse representations, we performed a comparison towards a baseline approach that was based on those dense embeddings. To ensure comparability, our baseline approach was based on the very same fasttext CBOW dense embeddings that we later created our sparse embeddings from.

Notice that the dense embeddings fit naturally into our framework, since utilizing the raw $m = 100$ dimensional dense embeddings can be viewed as performing (1) by choosing $k = m$, $\lambda = 0$ and $D \in \mathbb{R}^{k \times k}$ to be the identity matrix. Under these circumstances, the $\alpha = X$ is a trivial solution for (1), meaning that we are essentially using the original dense embeddings X. Applying our methodology involving the calculation of NPPMI based the raw dense embeddings, however, resulted in poor results.

In order to favor the application of dense embeddings, we made slight modifications in our framework when the inputs were dense emebddings. For the dense embeddings based baseline, we created a matrix $M \in \mathbb{R}^{8 \times 100}$, the rows of which contained unit normalized centroids for each class label that we obtained from averaging the term vectors that belong to each label. Upon making prediction for a dense embedding $\mathbf{v} \in \mathbb{R}^{100}$, we followed the same strategy as before, i.e. formed the product $\mathbf{v}M$ of the term vector and the matrix of unit normalized label centroids and took the argmax of the resulting vector. Table 1 includes the results of our baseline approach which was based on the centroids of the dense fasttext-CBOW embeddings.

Input	MA	AR
100d fasttext CBOW	78.0	1.35

Table 1: Baseline results for the label centroid-based approach using dense embeddings. MA and AR stands for mean accuracy and average ranking, respectively.

$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.4$	$\lambda = 0.5$
85.0	84.0	82.0	84.0	**86.0**

(a) Mean accuracy (MA)

$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.4$	$\lambda = 0.5$
1.33	1.34	1.33	1.33	**1.24**

(b) Average Rank (AR)

Table 2: Average Rank (AR) and Mean Accuracy (MA) metrics of models obtained for using different regularization coefficient λ when evaluated on the training data in a leave-one-out fashion using fasttext CBOW input embeddings and $k = 1000$.

4.2 Evaluation of our approach

Regarding the hyperparameters influencing our approach, we performed controlled experiments for analyzing the effects of changing the hyperparameter of both λ and k.

Controlling the regularization coefficient λ

We first performed controlled experiments to measure the effects of the regularization coefficient $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ while fixing the value of k to be 1000 following [Balogh *et al.*, 2020]. These results are included in Table 2. We can see that the choice for the regularization coefficient did not severely influence our evaluation scores.

By comparing the results in Table 2 with those in Table 1, we can see that our approach performs at least as good as the baseline approach which is based on the centroid of dense fasttext-CBOW embeddings. The contents of Table 1 demonstrate that the results obtained by relying on the sparse CBOW word representations were the best for the highest level of sparsity, i.e. when using $\lambda = 0.5$.

Jointly controlling the regularization and the dimensions

We subsequently measured the effects of simultaneously modifying the regularization coefficient λ and k, i.e. the number of basis vectors to be included in D. Figure 1 includes the results of those experimental settings for $(\lambda, k) \in (0, 100) \cup \{0.1, 0.2, 0.3, 0.4, 0.5\} \times \{1000, 1500, 2000\}$, i.e. we experimented with 15 different combinations of λ and k besides relying on the original 100-dimensional dense embeddings.

Figure 1 displays the MA and AR metrics along the x and y axis, respectively. We can see a negative correlation, i.e. the higher MA values we obtained the lower AR scores we registered. Since lower AR scores mean better performance this is a desired property of our approach. We can further notice that our approach produced substantially better results compared to the dense embeddings-based baseline. This is true for any combination of hyperparameters we tested our algorithms for and both for the MA and AR evaluation criteria.

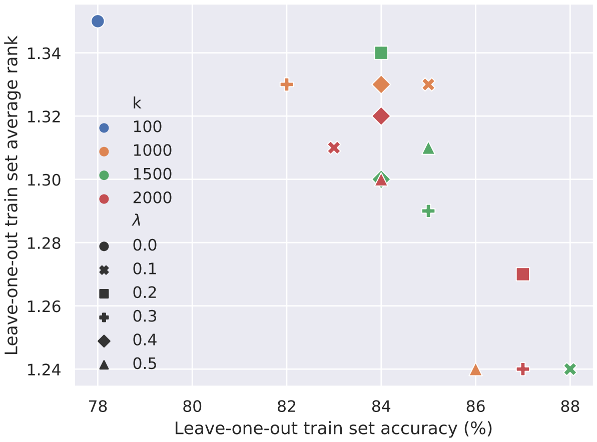


Figure 1: The joint effects of modifying the regularization coefficient λ and the number of basis vectors k when using 100-dimensional dense fasttext-CBOW embeddings as input. The performance of the centroid-based baseline is indicated by the blue dot in the upper-left corner of the scatter plot.

	Train (LOO)		Test	
Aggregation strategy	MA	AR	MA	AR
Ranking-based	**86.0**	**1.27**	**77.7**	**1.34**
Preceded by ℓ_2 normalization	85.0	1.30	74.7	1.37
Based on raw scores	85.0	1.30	73.7	1.38

Table 3: The effects of the different aggregation strategies when ensembling. The three aggregation strategies correspond to our three official submissions. Our official results are the ones labeled as Test.

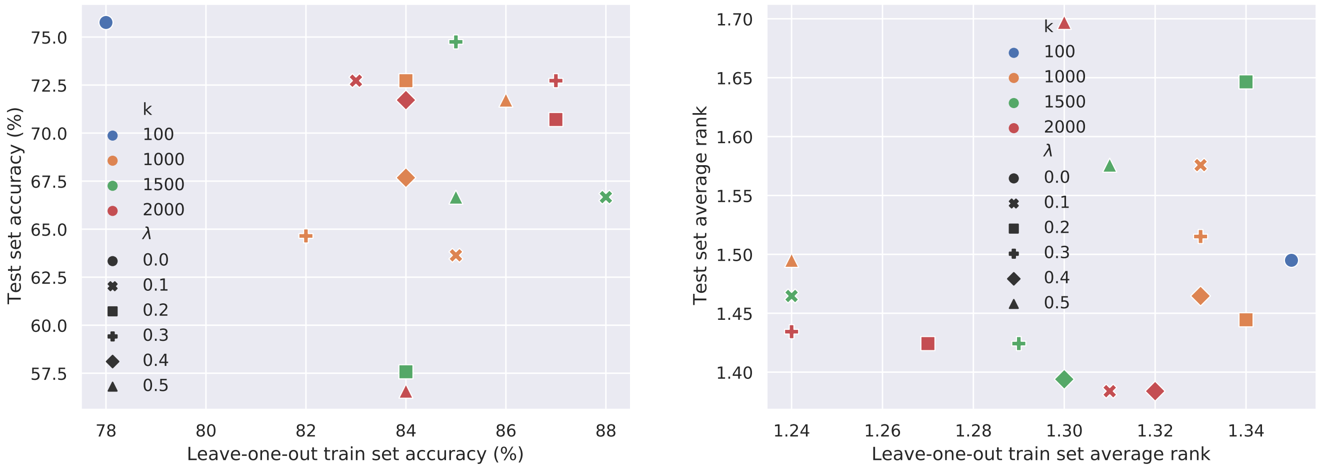
Taking an ensemble of models

In order to combine the independently constructed models that were obtained by different choices of hyperparameters, we derived our final predictions as a combination of the prediction of multiple models. We randomly chose 7 different models that were the result of different (k, λ) choices[2] and combined the predictions of these models.

We came up with three different ways of combining the predictions of the same independent models. The first approach only took into consideration the rankings that we obtained for each model but not the actual numerical scores of $\mathbf{s}^{(\mathbf{j})} = A^{(j)}\mathbf{v}$ with $\mathbf{s}^{(\mathbf{j})}$ denoting the association scores for the j^{th} model towards each class label.

The remaining two models differed in that they also considered the numeric scores for $\mathbf{s}^{(\mathbf{j})}$ upon combining them. One of the approaches that considered the actual numeric scores performed ℓ_2-normalization of the individual $\mathbf{s}^{(\mathbf{j})}$ vectors prior to summing them up, whereas the other alternative just summed up the raw scores in the distinct $\mathbf{s}^{(\mathbf{j})}$ vectors for making the final prediction.

[2] (1000, 0.4), (1000, 0.5), (1500, 0.3), (1500, 0.4), (2000, 0.1), (2000, 0.3), (2000, 0.5)



(a) Comparing the MA scores for the leave-one-out evaluation on the training data and the test set

(b) Comparing the AR scores for the leave-one-out evaluation on the training data and the test set

Figure 2: Systematic evaluation of selecting the various hyperparameters (k and λ) differently. The scatter plot includes the results of the MA and AR evaluations on the training set using leave-one-out evaluation and on the test set across the x any y axis, respectively. The $\lambda = 0$ ($k = 100$) case corresponds to the utilization of our dense embeddings-based baseline approach.

Table 3 includes the results of the ensemble models according to the three different ways of aggregating the $\mathrm{s}^{(j)}$ vectors for both the training terms in a leave-one-out manner and the test set. The results for the test set constitute the results of our official submission.

Our official results over the test set coincidentally resemble our leave-one-out evaluation scores obtained over the training set when employing our baseline approach which relies on the centroids of dense term embeddings (cf. the blue point in the upper-left corner of Figure 1) and the best test set results in bold included in Table 3).

Experiments with different input embeddings

After the gold labels for the test set of the shared task were released, we conducted a detailed experiment measuring the extent of different hyperparameter choices had similar effects when applying them on the training instances (in a leave-one-out fashion) and the test set. Figure 2 includes our comparison for all the combinations of λ and k when using the same fasttext-CBOW embeddings as before.

By looking at Figure 2, we can see that the relative performance of the dense embeddings based baseline is dominantly better on the test set when evaluated using MA as opposed to its performance over the training set. Interestingly, our baseline would even deliver the best performance on the test set in terms of MA, however, it would still offer a mediocre performance in terms of AR over the test set (cf. the blue points along the y axis in Figure 2). It is important to emphasize that the test set performance of our official submissions relying on an ensemble of sparse embeddings-based models outperforms that of the baseline approach for both evaluation metrics, i.e. it achieves a 77.7% MA (as opposed to 75.7% for our baseline) and a 1.34 AR (as opposed to 1.49 for the baseline).

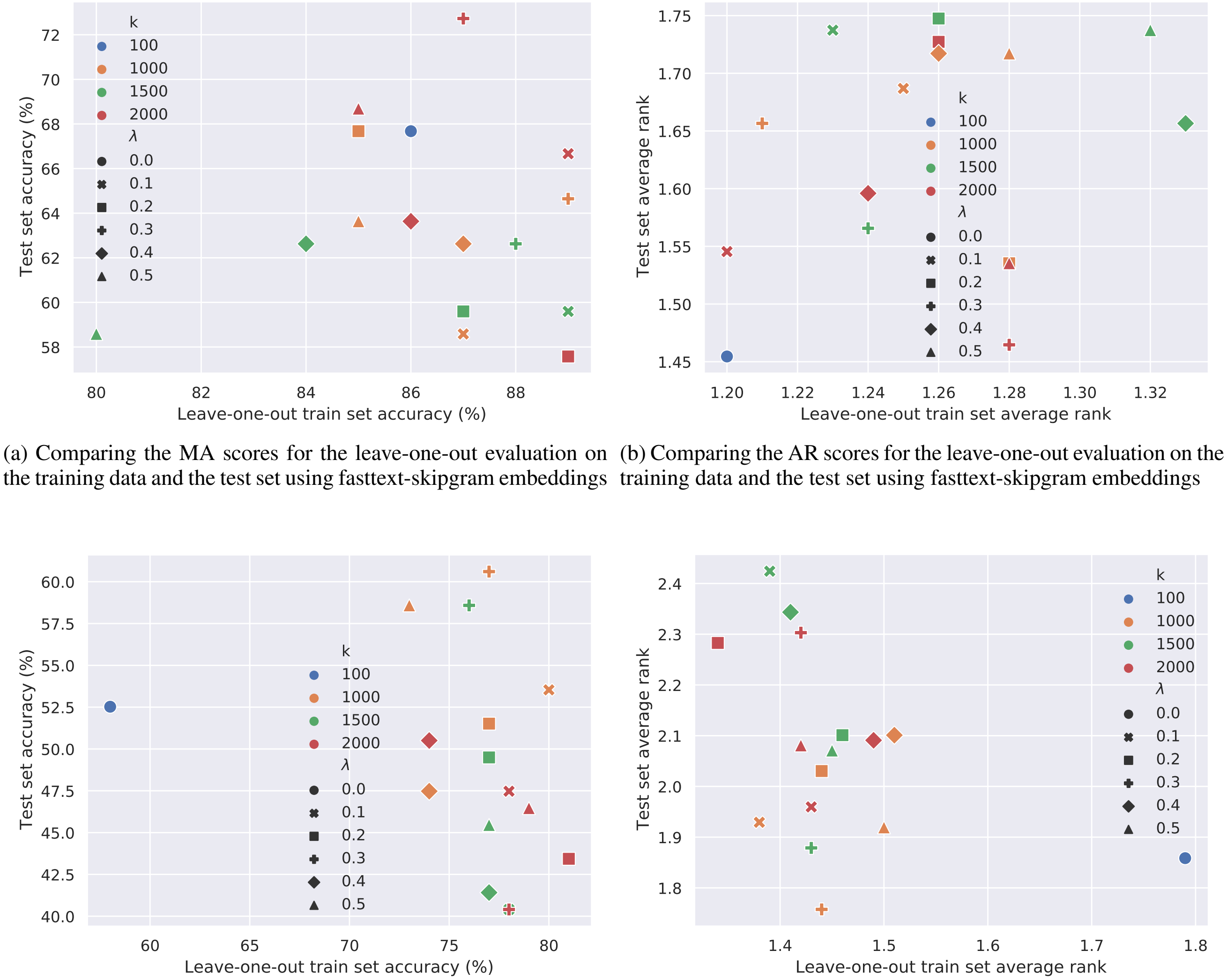
We next conducted similar experiments on alternatively trained dense embeddings. Besides the previously used fasttext-CBOW embeddings, we also trained fasttext-skipgram and Glove embeddings. Similar plots for the one in Figure 2 for these additional kinds of emebddings can be seen in Figure 3 for fasttext-skipgram (cf. Figure 3a and 3b) and Glove (cf. Figure 3c and 3d).

As illustrated in Figure 3, the dense fasttext-skipgram embedding baseline behaves complementary to what was seen for the fasttext-CBOW case, i.e. it yields the best AR performance, while not having outstanding capabilities in terms of MA. In summary, the best test set performance of the individual models based on fasttext-skipgram embeddings are 72.7% for MA (for $k = 2000, \lambda = 0.3$) and 1.45 for AR (for $k = 100, \lambda = 0$), none of which manages to surpass the performance of our ensemble model.

Looking at Figure 3, we can also conclude that Glove has the poorest performance on this task compared to any of the fasttext variants. Even the best MA scores delivered by Glove are around 80% and 60% when evaluating against the training and test set, respectively, whereas the fasttext variants are able to perform close to 90% and above 70% for the training and test sets, respectively.

5 Conclusions

In this paper we investigated the applicability of a general-purpose information theory-inspired algorithm for extracting ontological knowledge for the financial domain. Our experiments verified that by employing our algorithm allows us to predict ontological relations better as if we were relying on standard dense embeddings. Our source code for replicating our experiments is accessible from `https://github.com/begab/prosperAM-finsim`.



(a) Comparing the MA scores for the leave-one-out evaluation on the training data and the test set using fasttext-skipgram embeddings

(b) Comparing the AR scores for the leave-one-out evaluation on the training data and the test set using fasttext-skipgram embeddings

(c) Comparing the MA scores for the leave-one-out evaluation on the training data and the test set using Glove embeddings

(d) Comparing the AR scores for the leave-one-out evaluation on the training data and the test set using Glove embeddings

Figure 3: Systematic evaluation of selecting the various hyperparameters (k and λ) differently when employing fasttext-skipgram (3a, 3b) and Glove (3c, 3d). The scatter plot includes the results of the MA and AR evaluations on the training set using leave-one-out evaluation and on the test set across the x any y axis, respectively.

Acknowledgments

This research has been partly conducted in the project "Progressing Service Performance and Export Results of Advanced Manufacturers Networks", no CE1569 ProsperAMnet. The project has been supported by the European Fund for Regional Development in the framework of Interreg CENTRAL EUROPE 2019-2022. This research was also supported by the European Union and co-funded by the European Social Fund through the project "Integrated program for training new generation of scientists in the fields of computer science" (EFOP-3.6.3-VEKOP-16-2017-0002) and by the National Research, Development and Innovation Office of Hungary through the Artificial Intelligence National Excellence Program (2018-1.2.1-NKP-2018-00008).

References

[Balogh *et al.*, 2020] Vanda Balogh, Gábor Berend, Dimitrios I. Diochnos, and György Turán. Understanding the semantic content of sparse word embeddings using a commonsense knowledge base. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7399–7406, 2020.

[Berend *et al.*, 2018] Gábor Berend, Márton Makrai, and Péter Földiák. 300-sparsans at SemEval-2018 task 9: Hypernymy as interaction of sparse attributes. In *Proceedings* *of The 12th International Workshop on Semantic Evaluation*, pages 928–934, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[Berend, 2017] Gábor Berend. Sparse coding of neural word embeddings for multilingual sequence labeling. *Transactions of the Association for Computational Linguistics*, 5:247–261, 2017.

[Bernier-Colborne and Barrière, 2018] Gabriel Bernier-Colborne and Caroline Barrière. CRIM at SemEval-2018 task 9: A hybrid approach to hypernym discovery. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 725–731, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[Bojanowski *et al.*, 2017] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[Bouma, 2009] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of GSCL*, pages 31–40, 2009.

[Camacho-Collados *et al.*, 2018] Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa-Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. SemEval-2018 task 9: Hypernym discovery. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 712–724, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[Dash *et al.*, 2020] Sarthak Dash, Md. Faisal Mahbub Chowdhury, Alfio Gliozzo, Nandana Mihindukulasooriya, and Nicolas Rodolfo Fauceglia. Hypernym detection using strict partial order networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7626–7633, 2020.

[Faruqui *et al.*, 2015] Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1491–1500, July 2015.

[Hearst, 1992] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*, 1992.

[Held and Habash, 2019] William Held and Nizar Habash. The effectiveness of simple hybrid systems for hypernym discovery. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3362–3367, Florence, Italy, July 2019. Association for Computational Linguistics.

[Maarouf *et al.*, 2020] Ismail El Maarouf, Youness Mansar, Virginie Mouilleron, and Dialekti Valsamou-Stanislawski. The finsim 2020 shared task: Learning semantic representations for the financial domain. In *Proceedings of IJCAI-PRICAI 2020*, Kyoto, Japan (or virtual event), 2020.

[Mairal *et al.*, 2009] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, 2009.

[Murphy *et al.*, 2012] Brian Murphy, Partha Talukdar, and Tom Mitchell. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of COLING 2012*, pages 1933–1950, December 2012.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, October 2014.

[Speer and Havasi, 2012] Robert Speer and Catherine Havasi. Representing general relational knowledge in conceptnet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. European Language Resources Association (ELRA), 2012.

[Subramanian *et al.*, 2018] Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard H. Hovy. SPINE: sparse interpretable neural embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, pages 4921–4928, 2018.