

Evaluating Unsupervised Dutch Word Embeddings as a Linguistic Resource

Stéphan Tulkens, Chris Emmery, Walter Daelemans

CLiPS, University of Antwerp
firstname.lastname@uantwerpen.be

Abstract

Word embeddings have recently seen a strong increase in interest as a result of strong performance gains on a variety of tasks. However, most of this research also underlined the importance of benchmark datasets, and the difficulty of constructing these for a variety of language-specific tasks. Still, many of the datasets used in these tasks could prove to be fruitful linguistic resources, allowing for unique observations into language use and variability. In this paper we demonstrate the performance of multiple types of embeddings, created with both count and prediction-based architectures on a variety of corpora, in two language-specific tasks: relation evaluation, and dialect identification. For the latter, we compare unsupervised methods with a traditional, hand-crafted dictionary. With this research, we provide the embeddings themselves, the relation evaluation task benchmark for use in further research, and demonstrate how the benchmarked embeddings prove a useful unsupervised linguistic resource, effectively used in a downstream task.

Keywords: word embeddings, benchmarking, word2vec, SPPMI, language variation, dialect identification

1. Introduction

The strong variability of language use within, and across textual media (Collins et al., 1977; Linell, 1982) has on many occasions been marked as an important challenge for research in the area of computational linguistics (Resnik, 1999; Rosenfeld, 2000), in particular in applications to social media (Gouws et al., 2011). Formal and informal varieties, as well as an abundance of deviations from grammar and spelling conventions in the latter, drastically complicate computationally interpreting the meaning of, and relations between words. This task of understanding lies at the heart of natural language processing (NLP). Neural-network-based language models such as the models in `word2vec` have recently gained strong interest in NLP due to the fact that they improved state-of-the-art performance on a variety of tasks in the field. Given these developments, we found it surprising that only one set of word embeddings has been publicly released for Dutch (Al-Rfou et al., 2013), which does not offer sufficiently large dimensionality for state-of-the-art performance. The primary goal of this research is thus evaluating word embeddings derived from several popular Dutch corpora and the impact of these sources on their quality, specifically focusing on problems characteristic for Dutch. Word embeddings—being an unsupervised technique—cannot be easily evaluated without comparing performance in some downstream task. Therefore, we present two novel benchmarking tasks of our own making: a relation identification task analogous to previous evaluations on English, in which the quality of different kinds of word embeddings is measured, and a dialect identification task which measures the usefulness of word embeddings as a linguistic resource for Dutch in particular. In the literature, there has been some debate on the effectiveness of prediction-based embeddings when compared to more classical count-based embedding models (Baroni et al., 2014). As such, we train both count- (SPPMI) and prediction-based (SGNS) models, and compare them to previous efforts in both Dutch and English. Additionally,

we make the trained embeddings, the means to construct these models on new corpora, as well as the materials to evaluate their quality available to the research community¹.

2. Related Work

An idea mostly brought forward by the earlier distributional semantic models (DSMs), is that the context in which words occur (the distribution of the words surrounding them) can serve as a representation of their meaning, also known as the distributional hypothesis (Harris, 1954). Count based DSMs include LSA (Deerwester et al., 1989; Deerwester et al., 1990), PLSA (Hofmann, 1999) and LDA (Blei et al., 2003), which first create an explicit matrix of occurrence counts for a number of documents, and then factor this matrix into a low-dimensional, dense representation using Singular Value Decomposition (SVD) (Schütze and Silverstein, 1997). A more explicit way of implementing the distributional hypothesis is through the use of matrices containing co-occurrence counts (Lund and Burgess, 1996), which are then optionally transformed through the use of some information-theoretic measure, such as PMI (Pointwise Mutual Information) (Bullinaria and Levy, 2007; Levy and Goldberg, 2014) or entropy (Rohde et al., 2006). Over the years, these DSMs have proven adequate as a semantic representation in a variety of NLP tasks. An alternative to these count-based methods can be found in models predicting word identity from a given sentence context. Rather than deriving meaning from the representation of an entire corpus, these construct word representations one sentence at a time. In attempting to predict the current word through its context, the model will learn that words which occur in similar sentence contexts are semantically related. These representations are projected into n -dimensional vector spaces in which more similar words are closer together, and are therefore referred to as *word embeddings*. Recently, several models which create prediction-based word

¹Code and data are accessible via <https://github.com/clips/dutchembeddings>.

embeddings (Bengio et al., 2006; Collobert and Weston, 2008; Mnih and Hinton, 2009; Mikolov et al., 2013b; Pennington et al., 2014) have proved successful (Turian et al., 2010; Collobert et al., 2011; Baroni et al., 2014) and consequently have quickly found their way into many applications of NLP. Following Levy et al. (2014), we call the embeddings represented by dense vectors *implicit*, as it is not immediately clear what each dimension represents. Matrix-based sparse embeddings are then called *explicit* as each dimension represents a separate context, which is more easily interpretable. One of the more successful and most popular methods for creating word embeddings is `word2vec` (Mikolov et al., 2013a; Mikolov et al., 2013b). While `word2vec` often referred to as a single model, it is actually a collection of two different architectures, SkipGram (SG) and Continuous Bag of Words (CBoW), and two different training methods, hierarchical skipgram (HS) and negative sampling (NS). Levy et al. (2015) show that one of the architectures in the `word2vec` toolkit, SkipGram with Negative Sampling (SGNS) implicitly factorizes a co-occurrence matrix which has been shifted by a factor of $\log(k)$, where k is the number of negative samples. Negative samples, in this case, are noise words which do not belong to the context currently being modelled. Subsequently, the authors propose SPPMI, which is the explicit, count-based version of SGNS, i.e. it explicitly creates a co-occurrence matrix, and then shifts all cells in the matrix by $\log(k)$. SPPMI is therefore a count-based model which is theoretically equivalent to SGNS. When compared to other methods, such as GloVe (Pennington et al., 2014), SPPMI has showed increased performance (Levy et al., 2015).

3. Data

In our research, we used four large corpora, as well as a combination of three of these corpora to train both SPPMI and `word2vec`. Additionally, we retrieved a dataset of region-labeled Dutch social media posts, as well as hand-crafted dictionaries for the dialect identification task (see 4.3.).

3.1. Corpora

Roularta The Roularta corpus (Roularta Consortium, 2011) was compiled from a set of articles from the Belgian publishing consortium Roularta². Hence, the articles in this corpus display more characteristics of formal language than the other corpora.

Wikipedia We created a corpus of a Wikipedia dump³. The raw dump was then parsed using a Wikipedia parser, `wikiextractor`⁴, and tokenized using `Pattern` (De Smedt and Daelemans, 2012).

SoNaR The SoNaR corpus (Oostdijk et al., 2013) is compiled from a large number of disparate sources, including newsletters, press releases, books, magazines and newspapers. The SoNaR corpus (Nederlandse TaalUnie and

	Roul	Wiki	SoNaR	Comb	COW
# S	1.7m	24.8m	28.1m	54.8m	251.8m
# W	27.7m	392.0m	392.8m	803.0m	4b

Table 1: Sentence and word frequencies for the Roularta, Wikipedia, SoNaR500, Combined and COW corpora, where ‘m’ is million and ‘b’ billion.

STEVIN, 2014) therefore displays a high amount of variance in terms of word use and style. Unlike the COW corpus (see below), some spelling variation in the SoNaR corpus is automatically corrected, and the frequency of other languages in the corpus is reduced through the use of computational methods.

COW The COW corpus (Schäfer and Bildhauer, 2012) is a 4 billion word corpus which was automatically retrieved from domains from the .be and .nl top level domains in 2011 and 2014 (Schäfer and Bildhauer, 2012). As such, there is considerable language variability in the corpus. The corpus was automatically tokenized, although we did perform some extra pre-processing (see 3.2.).

Social Media Dataset The social media dataset was retrieved from several Dutch Facebook pages which all had the peculiarities of a specific dialect or province as their subject. As such, these pages contain a high percentage of dialect language utterances specific to that province or city. For each of these Facebook pages, the region of the page was determined, and all posts on these pages were then labelled as belonging to this region, resulting in a corpus of 96,000 posts. Tokenization and lemmatization of each post was performed using `Frog` (Bosch et al., 2007). This dataset is noisy in nature, and weakly labelled, as people might use standard language when talking about their province or home town, or will not use the ‘correct’ dialect on the designated page. This will prove the robustness of our models, and specifically that of our methods for ranking dialects.

Combined In addition to these corpora, we also created a Combined corpus, which consists of the concatenation of the Roularta, Wikipedia and SoNaR corpora, as described above. We created the Combined corpus to test whether adding more data would improve performance, and to observe whether the pattern of performance on our relation task would change as a result of the concatenation.

3.2. Preprocessing

Given that all corpora were already tokenized, all tokens were lowercased, and those solely consisting of non-alphanumeric characters were removed. Furthermore, sentences that were shorter than five tokens were removed, as these do not contain enough context words to provide meaningful results. Some additional preprocessing was performed on the COW corpus: as a side-effect of adapting the already tokenized version of the corpus, the Dutch section contains some incorrectly tokenized plurals, e.g. `regio's`, tokenized as `regi + o + ' + s`. Given this, we chose to remove all tokens that only consisted of

²<http://www.roularta.be/en>

³The 2015.07.03 dump, available at: <https://dumps.wikimedia.org/nlwiki/20150703>, retrieved on the 29th of July, 2015.

⁴<https://github.com/attardi/wikiextractor>

Province	ID	# words dict	# posts test
Antwerpen	ANT	10,108	20,340
Drenthe	-	1,308	0
Flevoland	-	1,794	0
Friesland	FRI	4,010	1,666
Gelderland	GEL	10,313	6,743
Groningen	GRO	7,843	147
Limburg	LI	45,337	10,259
Noord-Brabant	N-BR	20,380	1,979
Noord-Holland	N-HO	6,497	2,297
Oost-Vlaanderen	O-VL	23,947	14,494
Overijssel	-	4,138	0
Utrecht	UTR	1,130	7,672
Vlaams-Brabant	VL-BR	7,040	5,638
West-Vlaanderen	W-VL	16,031	12,344
Zeeland	ZEE	4,260	1,562
Zuid-Holland	Z-HO	6,374	11,221
Standard Dutch	-	133,768	-

Table 2: The type frequencies of the dialect dictionaries, the ID used in Figure 1, the type frequency of the corresponding dictionary, and the number of posts for that province in the test set.

one character, except the token `u`, which is a Dutch pronoun indicating politeness.

3.3. Dictionaries

To compare our embeddings to a hand-crafted linguistic resource, we collected a dictionary containing dialect words and sentences, as well as one for standard Dutch. The dialect dictionary was retrieved from MWB (Mijn Woordenboek)⁵, which offers user-submitted dialect words, sentences and sayings, and their translations. Only the dialect part was retained, and split in single words, which were then stored according to the region it was assigned to by MWB, and the province this region is part of. Any overlapping words across dialects were removed. As a reference dictionary for standard Dutch, the OpenTaal word list⁶ was used. Additionally, it was used to remove any general words from the dialect dictionaries, i.e. if a word occurred in both the Dutch reference dictionary and a dialect, it was deleted from the dialect. While employing hand-crafted dictionaries can be beneficial in many tasks, producing such resources is expensive, and often takes expert knowledge. Techniques able to use unlabelled data would not only avoid this, but could also prove to be more effective.

4. Experiments

For the evaluation of our Dutch word embeddings, we constructed both a novel benchmark task and downstream task, which can be used to evaluate the performance of new embeddings for Dutch.

⁵<http://www.mijnwoordenboek.nl/dialecten>, retrieved on 05/10/2015.

⁶Retrieved from <http://www.opentaal.org/bestanden> on 19/10/2015, version dated 24/08/2011

	#	Example	Translation
Superlative	26	‘slecht’ - ‘slechtst’	bad - worst
Past Tense	36	‘voorspellen’ - ‘voorspelde’	predict - predicted
Infinitive	29	‘dans’ - ‘dansen’	dance - danced
Comparative	35	‘groot’ - ‘groter’	big - bigger
Diminutive	29	‘spiegel’ - ‘spiegeltje’	mirror - small mirror
Plural	34	‘computer’ - ‘computers’	
Opposites	21	‘rationeel’ - ‘irrationeel’	rational - irrational
Currency	21	‘japan’ - ‘yen’	
Nationalities	29	‘belgië’ - ‘belg’	belgium - belgian
Country	48	‘noorwegen’ - ‘oslo’	norway - oslo
Gender	24	‘oom’ - ‘tante’	uncle - aunt

Table 3: Relation Evaluation set categories, examples, and translation of examples.

4.1. Parameter Estimation

For each corpus, we trained models using the `word2vec` implementation (Řehůřek and Sojka, 2010; Mikolov et al., 2013a) from `gensim`⁷. In order to determine optimal settings for the hyperparameters, several models were trained with different parameter values in parallel and were evaluated in the relation evaluation task (see below). For `word2vec` the SGNS architecture with a negative sampling of 15, a vector size of 320, and a window size of 11 maximized the quality across all corpora. For the SPPMI models, we created embeddings for the 50,000 most frequent words, experimenting with window sizes of 5 and 10, and shift constants of 1, 5 and 10. For all models, a shift constant of 1 and a window size of 5 produced the best results, the exception being the model based on the Roularta corpus, which performed best with a shift constant of 5 and a window size of 5. Relying on only one set of hyperparameters, as well as the performance of the relation task, could be seen as a point of contention. However, we argue in line with Schnabel et al. (2015) that ‘true’ performance across unrelated downstream tasks is complicated to assess. Nevertheless, we regard our approach to be satisfactory for the research presented here. Finally, in addition to our own models, we use the Polyglot embeddings (Al-Rfou et al., 2013) as a baseline, as this is currently the only available set of embeddings for Dutch.

4.2. Relation Identification

This task is based on the well-known relation identification dataset which was included with the original `word2vec` toolkit⁸, and which includes approximately 20,000 relation identification questions, each of the form: “If A has a relation to B, which word has the same relation to D?”. As such, it uses the fact that vectors are *compositional*. For example, given `man`, `woman`, and `king`, the answer to the question should be `queen`, the relation here being ‘gender’. In the original set, these questions were divided into several categories, some based on semantic relations, e.g. ‘opposites’ or ‘country capitals’, and some based on syntactic relations, e.g. ‘past tense’. Mirroring this, we created a similar evaluation set for Dutch. Considering the categories used, we aimed to replicate the original evaluation

⁷<https://radimrehurek.com/gensim/>

⁸see <https://code.google.com/archive/p/word2vec/>

	Polyglot	Wikipedia		Roularta		SoNaR500		Combined		COW	
		SPPMI	SGNS	SPPMI	SGNS	SPPMI	SGNS	SPPMI	SGNS	SPPMI	SGNS
Superlative	13.3	0.6	8.3	0.3	10.0	0.0	15.5	0.0	15.1	0	39.9
Past Tense	5.8	20.7	37.8	16.4	41.2	25.8	68.3	26.9	46.2	25	66.1
Infinitive	1.8	12.1	14.0	7.7	19.0	41.2	63.1	36.2	18.0	59	65.0
Comparative	18.6	12.1	39.0	17.6	43.8	41.2	63.4	40.0	55.5	53.7	76.6
Diminutive	10.0	3.6	5.8	0.0	3.1	2.1	20.7	1.7	10.1	14	44.9
Plural	3.8	44.4	36.2	20.9	10.5	34.9	37.5	42.2	43.9	57.4	56.1
Opposites	0.0	7.3	4.0	0.6	0.4	0.5	7.0	2.2	12.9	8.2	22.1
Currency	4.4	2.7	10.0	2.2	0.0	4.5	7.6	2.6	12.1	2.7	15.0
Nationalities	2.6	1.2	20.6	0.8	4.0	5.1	14.4	3.7	21.6	3.1	21.4
Country	1.9	20.2	47.1	2.2	2.8	14.3	36.6	22.8	52.1	25.1	43.1
Gender	25.3	30.6	52.9	25.2	21.9	44.7	75.9	45.1	64.9	50.7	72.5
Average	6.5	19.6	31.0	10.3	16.3	23.6	42.0	26.5	38.1	34.7	51.3

Table 4: Relation Identification set categories, the performance of the Polyglot baseline on this task, as well as that of SPPMI and SGNS trained on the listed corpora.

set as closely as possible, while also including some interesting syntactic phenomena in Dutch that are not present in English, such as the formation of diminutives. This resulted in 11 categories; 6 syntactic and 5 semantic. See Table 3 for an overview of the categories and an example for each category. Subsequently, we created a set of words occurring in all corpora—not taking into account word frequency—and retrieved applicable tuples of words from this vocabulary which fit the categories. By only taking words from the intersection of the vocabulary of all models, it is guaranteed that no model is unfairly penalized, assuring that every model is able to produce an embedding for each word in the evaluation set. After selecting approximately 20 tuples of words for each category, the 2-permutation of each set was taken separately, resulting in approximately 10,000 predicates.

As an evaluation measure the following procedure was performed for each set of embeddings: For each 2-permutation of tuples in the predicate evaluation set $(A, B), (C, D)$, where A, B, C , and D are distinct words, the following test was performed:

$$\arg \max_{v \in V} (\text{sim}(v, A - B + D)) \quad (1)$$

Where sim is the cosine similarity:

$$\text{sim}(w1, w2) = \frac{\vec{w1} \cdot \vec{w2}}{\|w1\| \|w2\|} \quad (2)$$

The objective is thus to find the word v in the vocabulary V which maximizes the similarity score with the vector $(A - B + D)$.

4.3. Dialect Identification

The relationship evaluation set above is a test of the quality of different embeddings. However, this does not prove the effectiveness of word embeddings as a linguistic resource. To counteract this, we created a task in which we try to detect dialectal variation in social media posts. The goal is to measure whether a resource that is equivalent to a hand-crafted resource can be created without any supervision. This identification of text containing dialect has been of interest to researchers across different languages such as

Spanish (Gonçalves and Sánchez, 2014), German (Scheffler et al., 2014), and Arabic (Lin et al., 2014). The task, then, is to correctly map text with dialect-specific language to the region of origin.

To test if the embeddings provide richer information regarding dialects than hand-crafted dictionaries, performance for both approaches needs to be compared. The amount of dialect groups for this task was determined based on the correspondence between those in the dialect dictionaries and a social media test set described in Section 3.3., which resulted in an identification task of at total 16 Dutch and Flemish provinces. For classification of dialect using the embeddings, we use each word in a document to rank the dialects for that document using two simple methods:

PROV using this method, we classify social media posts as belonging to a province by computing the similarity (as defined in Eq. 2) of every word in the post with all province names, and label the post with the province that was most similar to the highest amount of words. As such, we assume that the province which is most similar to a given word in n -dimensional space is the province to which that word belongs.

CO like PROV, but we also include countries, i.e. ‘Nederland’ and ‘België’ as possible targets. Hence, any words which are closer to either of the country names will not be assigned a province. This has a normalizing effect, as words from the general Dutch vocabulary will not get assigned a province.

We tested both these methods for SPPMI and SGNS models. For the dictionary the procedure was largely similar, but instead of distance a lookup through the dictionaries was used.

5. Results

5.1. Relation Identification

The results of the experiment on the relation identification are presented in Table 3, which shows that all models obtain higher performance on the syntactic categories when compared to the semantic categories, the exception being the ‘gender’ category, on which all models did comparatively

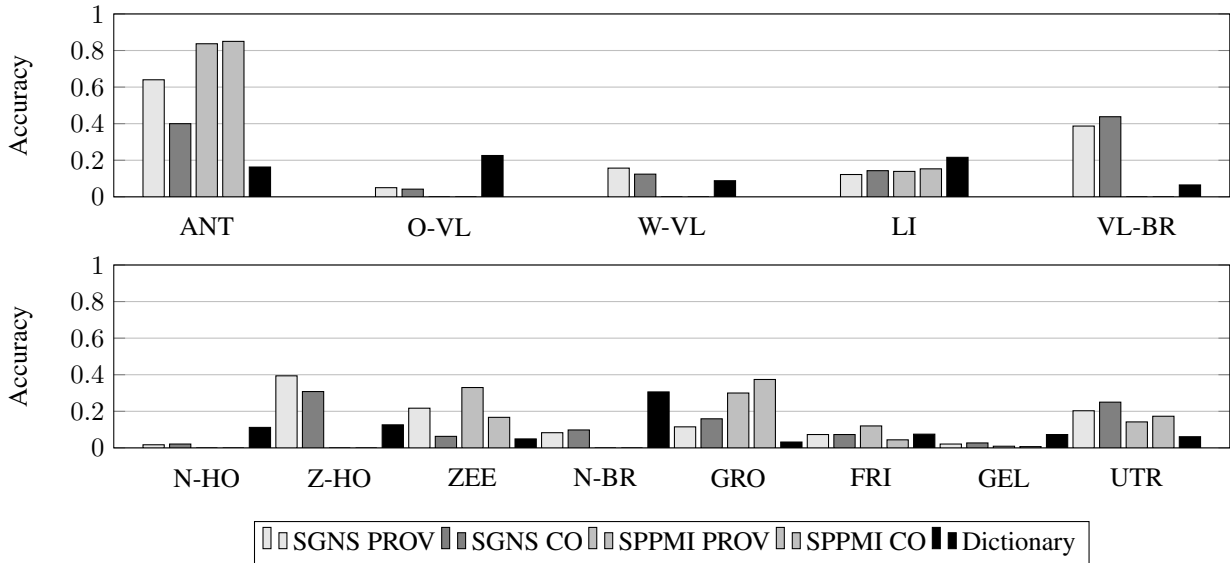


Figure 1: Accuracy scores per Flemish (top) and Dutch (bottom) province per model. Scores for the provinces of *Drenthe*, *Flevoland* and *Overijssel* are not listed, as these were not present in the test set.

	SGNS		SPPMI		DICT
	PROV	CO	PROV	CO	PROV
Acc	16.4%	13.6%	17.1%	17.8%	9.2%
MRR	27%	21.1%	22.1%	22%	14.3%

Table 5: Accuracy and MRR scores for SGNS, SPPMI, and the dictionary.

well. Furthermore, performance on ‘currency’ and ‘opposites’ was consistently low, the former of which could be explained through low occurrence of currencies in our data. All models outperform the baseline embeddings, which is made all the more problematic by the fact that the vocabulary of the baseline model was fairly small; only 6000 out of the 10,000 predicates were in vocabulary for the model. While it is not possible to estimate how the model would have performed on OOV (Out Of Vocabulary) words, this does demonstrate that our models perform well even given a large variety of words.

Comparing different SGNS models, it is safe to say that the biggest determiner of success is corpus size: the model based on the largest corpus obtains the highest score in 7 out of 11 categories, and is also the best scoring model overall. The Roularta embeddings, which are based on the smallest corpus, obtained the lowest score in 7 categories, and the lowest score overall. More interesting is the fact that the Combined corpus, does not manage to outperform the SoNaR corpus individually. This shows that combining corpora can cause interference, and diminish performance. Given the purported equivalence of SPPMI and SGNS, it is surprising that the performance of the SPPMI models was consistently lower than the performance of the SGNS models, although the SPPMI COW model did obtain the best performance on the plural category. None of the SPPMI models seem to be able to capture information about superlatives or nationalities reliably, with all scores for su-

perlatives close to 0 and, with the exception of the COW corpus, very low scores for nationality.

Finally, Mikolov et al. (2013a) report comparable performance (51.3 average) on the English variant of the relation dataset. While this does not reveal anything about the relative difficulty of the predicates in the dataset, it does show that our Dutch set yields comparable performance for a similar architecture.

5.2. Dialect Identification

As the models based on the COW corpus obtained the best results on the previous task, we used these in the dialect identification task. To determine the validity of using these models on our test data, we report coverage percentages for the models and dictionaries with regards to the test data vocabulary. The dialect part of our hand-crafted dictionaries had a coverage of 11.6%, which shows that the test set includes a large part of dialect words, as expected. The Dutch part of the dictionary covered 23.1% of the corpus. The SGNS model had a coverage of 68.3%, while the SPPMI model had a coverage of 24.4%, which is fairly low when compared to the SGNS model, but still more than either of the dictionaries in separation.

As our methods provide a ranking of provinces, both accuracy and mean reciprocal rank (MRR) were used to evaluate classification performance. While accuracy provides us with a fair measure of how well a dialect can be predicted for a downstream task, MRR can indicate if the correct dialect is still highly ranked. As summarized in Table 5, SPPMI obtained the highest accuracy score when countries were included as targets. When MRR was used as a metric, SGNS obtained the highest performance.

Performance per dialect is shown in Figure 1. Here, SGNS embeddings outperform the dictionaries in 7 out of 13 cases, and the SPPMI models outperform both the SGNS and dictionary models on several provinces. Regarding SPPMI, the figure reveals a more nuanced pattern of performance: for both tasks, the SPPMI model obtains surpris-

ingly high performance on the ANT dialect, while having good performance on several other dialects. This is offset, however, by the fact that the model attains a score of 0% on 6 provinces, and a very low score on 2 others. An explanation for this effect is that, being derived from a very large co-occurrence matrix, SPPMI is less able to generalize and more prone to frequency effects. To find support for this claim, we assessed the corpus frequencies of the province names in the COW corpus, and found that the names of all 6 provinces on which the SPPMI models obtained a score of 0 had a corpus frequency which was lower than 700. To illustrate; the name of the first high-frequent province, Overijssel, for which we do not have labeled data, has a frequency of 35218. Conversely, the provinces of Utrecht (UTR), Groningen (GRO), and Antwerpen (ANT) are all very high-frequent, and these are exactly the provinces on which the SPPMI model obtains comparably high performance. While the SGNS model showed a similar pattern of performance, it scored better on provinces whose names have a high corpus frequency, showing that it is influenced by frequency, but still is able to generalize beyond these frequency effects.

6. Conclusion

In this paper, we provided state-of-the-art word embeddings for Dutch derived from four corpora, comparing two different algorithms. Having high dimensionality, and being derived from large corpora, we hypothesized they were able to serve as a helpful resource in downstream tasks. To compare the efficiency of the embeddings and the algorithms used for deriving them, we performed two separate tasks: first, a relation identification task, highly similar to the relation identification task presented with the original `word2vec` toolkit, but adapted to specific phenomena present in the Dutch language. Here we showed to obtain better performance than the baseline model, comparable to that of the English `word2vec` results for this task. Secondly, a downstream dialect identification task, in which we showed that both methods we use for deriving word embeddings outperform expensive hand-crafted dialect resources using a simple unsupervised procedure.

7. Acknowledgements

We would like to thank TextGain for making the corpora of social media posts available to us, and Ákos Kádár for helpful remarks on our work. Part of this research was carried out in the framework of the Accumulate IWT SBO project, funded by the government agency for Innovation by Science and Technology (IWT).

8. Bibliographical References

Al-Rfou, R., Perozzi, B., and Skiena, S. (2013). Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria, August. Association for Computational Linguistics.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-

counting vs. context-predicting semantic vectors. In *ACL (1)*, pages 238–247.

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., and Gauvain, J.-L. (2006). Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Bosch, A. v. d., Busser, B., Canisius, S., and Daelemans, W. (2007). An efficient memory-based morphosyntactic tagger and parser for dutch. *LOT Occasional Series*, 7:191–206.

Bullinaria, J. A. and Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39(3):510–526.

Collins, A., Brown, J. S., and Larkin, K. M. (1977). Inference in text understanding. *BBN report; no. 3684*.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

De Smedt, T. and Daelemans, W. (2012). Pattern for python. *The Journal of Machine Learning Research*, 13(1):2063–2067.

Deerwester, S. C., Dumais, S. T., Furnas, G. W., Harshman, R. A., Landauer, T. K., Lochbaum, K. E., and Streeter, L. A. (1989). Computer information retrieval using latent semantic structure, June 13. US Patent 4,839,853.

Deerwester, S. C., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391.

Gonçalves, B. and Sánchez, D. (2014). Crowdsourcing dialect characterization through twitter. *PLoS one*, 9(11):e112074.

Gouws, S., Metzler, D., Cai, C., and Hovy, E. (2011). Contextual bearing on linguistic variation in social media. In *Proceedings of the Workshop on Languages in Social Media*, pages 20–29. Association for Computational Linguistics.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM.

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185.

Levy, O., Goldberg, Y., and Ramat-Gan, I. (2014). Linguistic regularities in sparse and explicit word representations. *CoNLL-2014*, page 171.

- Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.
- Lin, C.-C., Ammar, W., Levin, L., and Dyer, C. (2014). The cmu submission for the shared task on language identification in code-switched data. *EMNLP 2014*, page 80.
- Linell, P. (1982). *The Written Language Bias in Linguistics*. Linköping University Electronic Press.
- Lund, K. and Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Mnih, A. and Hinton, G. E. (2009). A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088.
- Oostdijk, N., Reynaert, M., Hoste, V., and Schuurman, I. (2013). The construction of a 500-million-word reference corpus of contemporary written dutch. In *Essential speech and language technology for Dutch*, pages 219–247. Springer.
- Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.
- Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. <http://is.muni.cz/publication/884893/en>.
- Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res.(JAIR)*, 11:95–130.
- Rohde, D. L., Gonnerman, L. M., and Plaut, D. C. (2006). An improved model of semantic similarity based on lexical co-occurrence. *Communications of the ACM*, 8:627–633.
- Rosenfeld, R. (2000). Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278.
- Schäfer, R. and Bildhauer, F. (2012). Building large corpora from the web using a new efficient tool chain. In *LREC*, pages 486–493.
- Scheffler, T., Gontrum, J., Wegel, M., and Wendler, S. (2014). Mapping german tweets to geographic regions. In *Proceedings of the NLP4CMC Workshop at Konvens*.
- Schütze, H. and Silverstein, C. (1997). Projections for efficient document clustering. In *ACM SIGIR Forum*, volume 31, pages 74–81. ACM.
- Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

9. Language Resource References

- Nederlandse TaalUnie and STEVIN. (2014). *SoNaR500*. Nederlandse TaalUnie, Corpora On the Web, ISLRN 613-489-674-355-0.
- Roularta Consortium. (2011). *Roularta Corpus*. Roularta Consortium.
- Schäfer, R. and Bildhauer, F. (2012). Building large corpora from the web using a new efficient tool chain. In *LREC*, pages 486–493.