Irish Communication Review

Volume 16 | Issue 1 Article 6

June 2018

Text Analytics Techniques in the Digital World: Word Embeddings and Bias

Marisa Llorens
Technological University Dublin, marisa.llorens@tudublin.ie

Text analytics techniques in the digital world: Word embeddings and bias

Marisa Llorens Salvador

Abstract

The proliferation of textual data in the form of online news articles and social media feeds has had an impact on the text analytics developments in recent years. Some of the challenges of natural language processing, understanding and generation have been successfully resolved and the results are applications such as AI personal assistants and bots. Word embeddings are an example of these successful solutions where unsupervised data-driven algorithms are used to understand concepts and relationships between words. This paper presents a description of word embedding algorithms, and a discussion on how bias in the training data can be captured, reproduced and even amplified by the algorithms.

Introduction

The task of understanding natural language is considered a hard task in the area of computer science. The fields of natural language processing (NLP) and artificial intelligence (AI) have been successful in small language tasks such as finding similarities and associations between words; however, a human-like understanding of language remains an unsolved challenge.

Natural language processing techniques and AI algorithms are used to transform human (natural) languages into a set of features that can then be used to perform tasks aimed at understanding or creating human language.

From an AI point of view two differentiated approaches can be taken (i) unsupervised and (ii) supervised algorithms.

In the first approach, an unsupervised algorithm is used to find relationships between inputs. In this case, the algorithm looks at the input data, modelling and understanding the structure by finding similar characteristics. Similar inputs can then be grouped together in clusters, abstracting them and giving a name (label) to each grouping. Clustering is a compression of information processing: similar inputs are collected together in an intelligent and data-driven manner. When similar inputs are grouped together, the analysis of the data can be performed on group meanings instead of individual inputs, hence obtaining higher levels of abstraction and reducing the conceptual load.

A supervised algorithm, on the other hand, uses examples to build an association between the input and the output. The examples are called training data and include a set of features and a label for each input. The association created between the input and the output is a flexible model containing tuneable parameters. The parameters are fixed during the training phase by forcing that the learned model works correctly for the labelled examples in the training set.

In both cases, the algorithm's learning process is based on the data available, and hence the general term of machine learning or data driven learning. The main conceptual difference between the supervised and unsupervised approach is the use of labels (human input is required to create the labels) in the training phase that *guide* the learning process, whereas the unsupervised algorithms find relationships within the input data without the need of labels or *guided* learning.

The concept of word embedding

Word embeddings (Mikolov et al., 2013) are unsupervised techniques used to map words or phrases from a text to a corresponding vector of real numbers. This representation involves building a low dimensional continuous vector space from a high dimensional space (one dimension per word). The obtained vector space preserves the contextual similarity of words – therefore words that appear regularly together in text will also appear together in the vector space.

The idea that words with similar meanings appear in similar contexts is called the distributional hypothesis, and the models that follow this idea are called distributional semantic models. These models provide a framework to compute semantic relationship between words. Semantic distributional models such as Latent Semantic Analysis (LSA) have been popular since the 1990s and have shown strong performances in finding word similarities. LSA models are counter-based models; word embeddings, meanwhile, are predictive models. Predictive models have shown strong performances when using large data sets whereas LSA models have limitations due to their high memory use for large datasets. In cases where the data is scarce, LSA models have been found to perform better then predictive models (Altszyler et al., 2016). The use of co-occurrences in a counter-based model leads to a high dimensional space where each word of the vocabulary is represented in an array of co-occurrences.

For example, the co-occurrences of word1 and each of the other words in the document can be represented in an array:

$$word1 = [w1w2, w1w3, w1w4, \ldots\ldots w1wn]$$

where *w1wn* represents the number of times words 1 and n appear together in a document.

The dimensions of the vectors depend on the size of the vocabulary, obtaining large vectors; whereas in the word embeddings model, values in the array attempt to represent concepts and meanings, hence reducing the dimension of the vectors obtained. For example, different representations of capital cities and countries can capture different relationships between capital cities and their country. Sample values for two different representations: (i) co-occurrences of words and (ii) word embeddings are used in Table 1 to illustrate the different relationships and how the representations capture information and meaning.

Co-occurrences Vector size = n	Word embedding Vector size < n
Madrid=[**1.6**, 0, 0, **0.1**, 0, 0,..,0]	Madrid=[**0.91**, **0.84**, 0.1, 0…., **0.42**]
London=[0, **1.1**, **0.2**, 0, 0, 0,…,0]	London=[**0.93**, **0.81**, 0.11, 0,…, **0.96**]
UK=[0, **1.7**, **0.5**, 0, 0, 0, 0, 0…,0]	UK=[**0.32**, **0.74**, 0.6, 0,…, **0.97**]

Spain=[**1.1**, 0, 0, **0.12**, 0, 0, …,0]	Spain=[**0.31**, **0.71**, 0.64, 0,…, **0.41**]

Table 1 Co-occurrences vs. word embedding representations

The values contained in Table 1 show a direct relation between the pairs Madrid-Spain and London-UK with similar vectors of co-occurrences. These relationships are captured from texts when both words appear together frequently in the same sentence or document. However, in the vector model space created using word embeddings, additional relationships between concepts can also be found. For example, the concepts of country and capital city and how they relate to each other can be seen in the similarities between UK-Spain and London-Madrid (first elements of vectors on Table 2).

Capitals	Countries
Madrid=[**0.91**, **0.84,…..]**	UK=[**0.32**, **0.74**, …..]
London=[**0.93**, **0.81**, …..]	Spain=[**0.31**, **0.71**, …..]

Table 2 Vector values showing relationships between capital cities and between countries

Furthermore, the values at the end of the vector connect each capital city with its corresponding country (Table 3).

Capital – Country Relationship	Capital – Country Relationship
Madrid=[**….., 0.42**]	London=[…, **0.96**]
Spain=[**….., 0.41**]	UK=[…, **0.97**]

Table 3 Vector values showing capital country relationship

The conceptual connections created in the vector space allow for vector calculations to be performed on the data. Word embedding algorithms assign similar vectors to similar words, hence placing them in the same area of the vector space. The distances between capital city and country are similar in both cases, allowing for mathematical vector calculations, such as addition and subtraction.

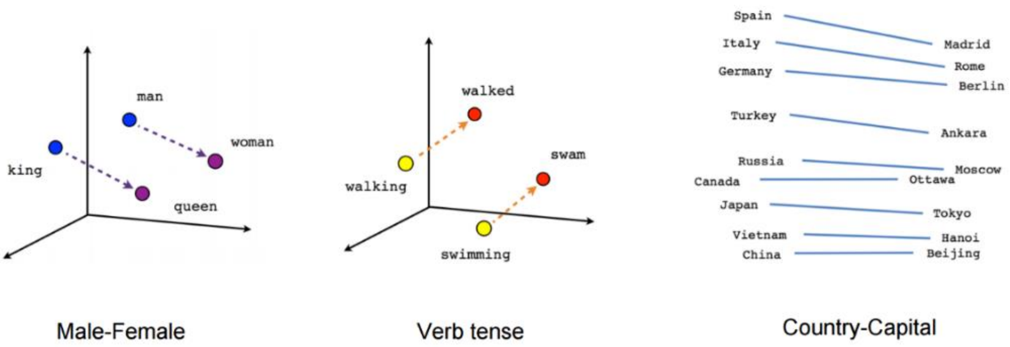

Figure 1 Vector space[1]

The word representations are capable of capturing both semantic and syntactic regularities. These regularities appear as similar offset distances between pairs of words sharing a particular relationship.

Examples of these relationships and results for vector calculations are:

Vector London – Vector UK + Vector Spain = Madrid

Vector King – Vector man + Vector woman = queen

Vector walked – Vector walking + Vector swimming = swam

This capacity of performing vector calculations provides the model with what can be argued to mean a certain level of conceptual *understanding*.

The word embedding algorithms

Vector space models have been used in NLP applications since the 1990s. However in recent years a new set of tools, such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), have brought word embeddings to the forefront of NLP research. These models are a successful implementation of unsupervised learning and can be either custom trained or distributed as pre-trained embeddings.

One of the most popular word embedding models is word2vec. The main characteristics of word2vec are the production of useful word representations, its efficient training process and its scalability to large word and corpora vocabularies (Levy & Goldberg, 2014). In terms of word2vec implementations, two different

[1] https://www.tensorflow.org/tutorials/word2vec

models are commonly used: (i) continuous bags of words model (CBOW) and (ii) the *skip-gram with negative sampling* model (Mikolov et al., 2013).

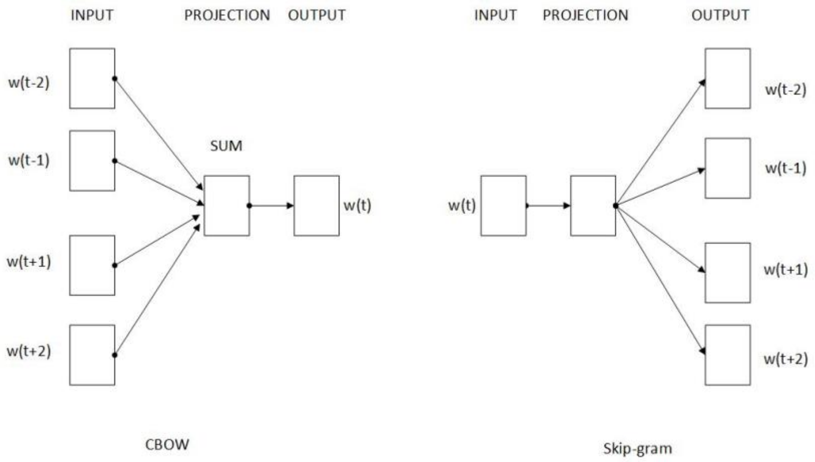

Figure 2 Continuous bags of Words and Skip-gram models

Both CBOW and skip-grams are based on an artificial neural network architecture that has been modified to eliminate the non-linear hidden layer and the projected layer is shared for all words. The CBOW model is called continuous bag-of-words as the order of the words does not influence the projection. For each target word w_t the model receives a window of *n* words, situated around it, at each time step *t*. The algorithm then predicts the current word, based on the context words.

The architecture of the skip-grams model is similar to the CBOW model; however, instead of predicting the current word based on the context words, it uses the current word as an input to a classifier with continuous projection layer and it predicts words within a certain range before and after the current word. Words appearing closer to the current word are given higher weights by sampling more nearby words and less distant words in the training examples.

The natural language processing research group in Stanford University created in 2014 their own word embedding algorithm, Global Vectors for word representation, GloVe (Pennington et al., 2014). GloVe uses global statistical methods to identify the underlying co-occurrence statistics of the corpus while also capturing the linear substructures observed in prediction-based methods like word2vec. Co-occurrence probabilities for target words (i, j) and context words (k) as well as their ratios $(Pik/Pjk\ )$ are used to calculate vector distances and find conceptual relationships.

Both word2vec and GloVe have shown success in word similarity and word analogy tasks as well as at name entity recognition, which involves recognizing names that identify entities such as persons, locations or organizations for example (Ivanitskiy et al., 2016; Sienčnik, 2015) .

Uses of text analytics and word embeddings

A large amount of the information generated by humans is in text format or language related, making the analysis of textual data an important area in the data analytics field. Furthermore, the generation of language by artificial entities is an aspiration in the area of AI from its early days. In the context of today's hyper connected society, assistive technologies, access to 24-hour customer services bots and artificial personal assistants, such as Apple's Siri and Amazon's Alexa, are some of the successful applications of natural language processing and generation which use word embeddings.

Other examples of text analytic applications that use word embeddings are machine translation (Zou et al., 2013), suggested search terms and results for web search engines (Mitra et al., 2017; Nie et al., 2017), news summary generation, social media information mining (Nikfarjam et al, 2015), sentiment analysis (Severyn & Moschitti, 2015; Tang et al., 2014), HR preselection algorithms (Tosik et al., 2015) and criminal risk profiler for court sentencing, decision on bail and parole.

The number of applications keeps expanding every day with tech companies finding new niche areas where text analytics can offer solutions to automate tasks.

Bias issues

The different word embedding models described in this paper, as well as other types of unsupervised algorithms, are instances of data led model generation. In these models, a large amount of data is used to train the model and get it to 'learn' particular patterns. These patterns can be language related in the case of models trained to generate text or they can be numerical in the case of, for example, energy network analysis and prediction. In both situations, historical data is used to create the 'knowledge' (trained model) with predictive capabilities.

In many circumstances, data led algorithms outperform other systems that require expert knowledge and complex probability calculations to predict outcomes for future situations. Data led algorithms can be trained on different data to obtain a different model, making these algorithms multipurpose.

However, the use of unsupervised algorithms trained on user-generated data poses the risk of reproducing the bias present in the data. Female/male gender stereotypes have appeared on word embeddings trained on Google News data (Bolukbasi et al., 2016). The obtained word embeddings connect *'queen'* to '*woman'* the same way they connect '*receptionist'* to '*woman'*. The existence of genderless nouns in English can be used to analyse stereotypes by looking at the associations between those nouns and the words he and she. For example, the following equality has been observed (Bolukbasi et al., 2016) in word vectors trained by GloVe:

$$vect(man) - vect(woman) \approx vect(programmer) - \ vect(homemaker)$$

The projection of words along the she-he axis offers a graphical representation of related concepts and it can be used to visualize gender bias in the distribution. Focussing on genderless nouns in English to describe professions, it can be found that words such as politician, brilliant, arrogant, architect and great appear on the 'male' side of the graph , whereas mom, housewife, fiancée, girlfriend, diva, princess and uterus appear on the 'female' side of the graph (Bolukbasi et al., 2016).

In this paper, a similar projection of words along the x axis and using GloVe embeddings pre-trained using Wikipedia articles was calculated. A simple vector calculation using the 30th first related terms for $vector(man) + vector(science)$ and $vector(woman) + vector(science)$ obtained can be seen in Figure 3. This figure shows a bias in the disciplines associated with the different genders. On the male side of the graph, disciplines like physics, philosophy and mathematics appear, whereas arts, writing and literature appear on the female side of the projection. Generic terms such as graduate, research, studies, teaching, education and others appear on both sides of the graph.

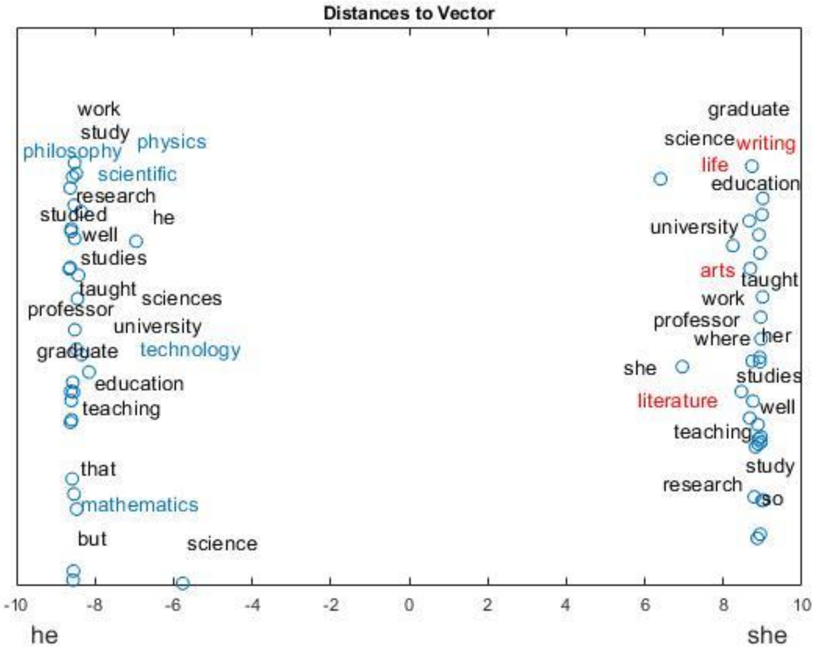

Figure 3 Man and woman + vector(science) using GloVe pretrained embeddings

Other biases found in the literature are European American and African American typical names associated with pleasant and unpleasant attributes (Caliskan et al., 2017). The results of this research indicate that known historical biases (Greenwald, 2017; Greenwald et al., 1998) are captured and reproduced by machine learning algorithms.

Different word embedding algorithms learn concepts from data in slight different ways depending on the details of their implementation; however, the performance and results of the different models are similar when using the same data. It is the data fed in to the algorithm that will determine the differences in the outcomes. In other words, different algorithms learning using the same data obtain similar results. Large amounts of data (datasets in the region of billions of words per dataset) such as Wikipedia articles (the full site), Google News and Twitter messages are used as input data in an effort to feed the algorithms with large amounts of real language data. This dependency on large amounts of data and the effect this data has on the results obtained pose a challenge for researchers in the quest for unbiased natural language results.

The effect of the input data in the resulting word embeddings can be observed in the embeddings obtained for the same vector calculations using different training data. In this experiment, two different pre-trained models were used: (i) Wikipedia data and (ii) Twitter data. Table 2 shows a sample of the results obtained for *man + abortion* and *woman + abortion* using a model trained on Wikipedia and Twitter data respectively.

After stop words and words that add no meaning have been eliminated the results show that, in both cases, *man + abortion* is associated with the word *right*, whereas *woman + abortion* using Wikipedia results in the word *victim* and using Twitter results in the word *murder*.

Wikipedia	Twitter
Man + abortion = right	Man + abortion = right
Woman + abortion = victim	Woman + abortion = murder

Table 4 Wikipedia and Twitter vector calculations

Hard and soft debiasing are the main two approaches used to avoid bias in word embedding results. Hard debiasing uses human inputs to identify bias whereas soft debiasing focusses mainly on the algorithmic computation of the word embeddings. Hard debiasing shows better results both reducing the bias but also preserving useful gender relationships (Bolukbasi et al., 2016).

The resulting word embeddings not only capture historical and cultural biases, they can also amplify the inherent original bias of the training data. For example, researchers have studied the biases found in image recognition where the training data is a set of images with captions (Zhao et al., 2017). The algorithm aims to create an automated caption for new pictures. This research shows that biases contained in the training data such as ‘cooking’ being 33% more likely to involve females than males, can be amplified to 68% at test time.

The combination of large amounts of decentralized, user-generated data with unsupervised algorithms that find hidden patterns in that data can lead to biased results in two ways. In the first instance, it can reproduce biases contained in the original data. In the second instance, the algorithms can enhance previous biases by identifying biased parameters as fundamental characteristics of the concept.

The use of biased word embeddings in different applications pose a threat to fair decision making processes as the inherent bias is automatically passed on to any application that uses the word embeddings, perpetuating in this way cultural stereotypes. For example, the search "computer programmer cv" may result in male applicants being ranked higher than female applicants, whereas the opposite effect may happen when searching through midwife cvs.

Conclusions

The use of unsupervised word embedding algorithms has proven to be a successful application of natural language processing and generation. Word embeddings use large amounts of information (training data) to identify concepts and conceptual relationships between words. These concepts are stored using vectors and vector space calculations can be used to explore new relationships between words.

In order to obtain a powerful word embedding model, large datasets of textual data have to be used to train the algorithm. The datasets used for generating the models are typically Wikipedia, Google News or Twitter, given the availability and size of the data. Using these datasets researchers aim at capturing language use and find relationships between concepts. However, these relationships can be tainted by historical and cultural biases found in the data. For example, looking at political data prior to 1960, *prime minister* is a title that an algorithm would identify as a male only title as until 1960 no woman had ever held that title (Sirimavo Bandaranaike, prime minister of Ceylon and Sri Lanka 1960).

Different word embedding algorithms have been developed in recent times, two of the most popular ones being word2vec and GloVe. These algorithms have been found to produce similar results when trained with the same data, whereas different training data on the same algorithm produces different results, showing a dependency on the training data – this is not a surprise, as this is a characteristic of all data driven algorithms.

However, two main issues arise when discussing word embedding results: inherent bias in the training data and amplification of bias.

The use of uncontrolled user-generated data such as Twitter feeds provides large amounts of up to date textual information; however, it poses a risk as bias can be present in the training data and reproduced in the resulting word embeddings.

This bias can be historical or cultural but it can also be ideological and used deliberately to affect the unsupervised models.

Furthermore, the word embedding algorithms can also amplify the bias contained in the original training data as they are designed to generalize models using the information contained in the original data. This can lead to higher weights to be allocated to biased parameters.

The results of word embedding algorithms affect how we relate to the world in web search results, language generation applications such as customer service or social media bots and news summaries generation. A double amplification effect can occur: in the first instance, the bias from the training data is amplified after going through the algorithm and this bias later contributes to the bias of society. Individuals in this society generate the training data available for training the algorithms. Given the unsupervised nature of the algorithms and the direct proportionality between their power and the amount of data used to train them, the uncontrolled use of data led algorithms such as word embeddings can lead to a spiral of bias amplification.

While the scientific community works on robust algorithms immune to bias and society works towards eliminating historical and cultural biases, a general understanding of the unsupervised algorithmic process, as well as traceability of data used in model generation, should be advisable for any user having interactions with natural language applications.

References

Altszyler, E., Sigman, M., & Slezak, D. F. (2016). Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database. Science, 8.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker?: debiasing word embeddings. In Advances in Neural Information Processing Systems (pp. 4349–4357).

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. Science, 356, 183–186.

Greenwald, A. G. (2017). An AI stereotype catcher. Science, 356, 133–134.

Greenwald, A. G., McGhee, D. E., & Schwartz, J. L. (1998). Measuring individual differences in implicit cognition: the implicit association test. Journal of Personality and Social Psychology, 74.

Ivanitskiy, R., Shipilo, A., & Kovriguina, L. (2016). Russian Named Entities Recognition and Classification Using Distributed Word and Phrase Representations. In SIMBig (pp. 150–156).

Levy, O., & Goldberg, Y. (2014). Dependency-based word embeddings. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (Vol. 2, pp. 302–308).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.764.2227

Mitra, B., Diaz, F., & Craswell, N. (2017). Learning to match using local and distributed representations of text for web search. In International World Wide Web Conferences Steering Committee, Proceedings of the 26th International Conference on World Wide Web (pp. 1291–1299).

Nie, T., Ding, Y., Zhao, C., Lin, Y., Utsuro, T., & Kawada, Y. (2017). Clustering search engine suggests by integrating a topic model and word embeddings. In IEEE, Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2017 18th IEEE/ACIS International Conference on (pp. 581–586).

Nikfarjam, A., Sarker, A., O'Connor, K., Ginn, R., & Gonzalez, G. (2015). Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. Journal of the American Medical Informatics Association, 22, 671–681.

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

Severyn, A., & Moschitti, A. (2015). Twitter sentiment analysis with deep convolutional neural networks. In ACM, Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 959–962).

Sienčnik, S. K. (2015). Adapting word2vec to named entity recognition. In Linköping University Electronic Press, Proceedings of the 20th nordic conference of computational linguistics, nodalida 2015, may 11-13, 2015, vilnius, lithuania (pp. 239–243).

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014). Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics

(Volume 1: Long Papers) (pp. 1555–1565). Baltimore, Maryland: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/P14-1146

Tosik, M., Hansen, C. L., Goossen, G., & Rotaru, M. (2015). Word embeddings vs word types for sequence labeling: the curious case of CV parsing. In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing (pp. 123–128).

Zhao, H., Du, L., & Buntine, W. (2017). A Word Embeddings Informed Focused Topic Model. In Asian Conference on Machine Learning (pp. 423–438).

Zou, W. Y., Socher, R., Cer, D., & Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1393–1398).