Proceedings of the Federated Conference on Computer Science and Information Systems pp. 361–365
DOI: 10.15439/2017F204
ISSN 2300-5963 ACSIS, Vol. 11

A Hierarchical Approach for Sentiment Analysis and Categorization of Turkish Written Customer Relationship Management Data

Mehmet Saygın Seyfioğlu
Cyber Security and Big Data Department
STM Defense Technologies Engineering and Trade Corp
TOBB University of Economics and Technology
Ankara, Turkey
Email: msaygin.seyfioglu@stm.com.tr

Mustafa Umut Demirezen
Cyber Security and Big Data Department
STM Defense Technologies Engineering and Trade Corp
Ankara, Turkey
Email: udemirezen@stm.com.tr

***Abstract*—Today, large scale companies are receiving tens of thousands of feedback from their customers every day, which makes it impossible for them to evaluate the feedbacks manually. As sentiments expressed by the customers are vitally important for companies, an accurate and swift analysis is needed. In this paper, a hierarchical approach is proposed for sentiment analysis and further categorization of Turkish written customer feedback to a private airline company. First, the word embeddings of customer feedbacks are computed by using Word2Vec then averaged in proportion with the inverse of their frequency in the document. For binary sentiment analysis, i.e determination of 'positive' and 'negative' sentiments, an extreme gradient boosting (xgboost) classifier is trained on averaged review vectors and an overall accuracy of 92.5% is obtained which is 16.8% higher than that of the baseline model. For further categorization of negative sentiments in one of twelve pre determined classes, an xgboost classifier is trained upon document embeddings of negatively classified comments, which were calculated using Doc2Vec. An overall accuracy of 71.16% is obtained for the task of categorization of 12 different classes using the Doc2Vec approach, thereby yielding a classification accuracy 19.1% higher than that of the baseline model.**

***Index Terms*—customer relationship management, word2vec, doc2vec, classification, sentiment analysis, xgboost**

I. Introduction

CUSTOMER Relationship Management (CRM) has gained importance with the advent of the big data phenomenon. Millions of customers are sharing their opinions about the products they use every day. According to [1], 77% of customers care about other people's comments, while 75% of customers trust comments on social media rather than personal recommendations. CRM enables companies to focus on their customers' needs: e.g., what do they want and what needs to be fixed [2]. When a problem occurs, swift action needs to be taken by companies according to customer feedback to prevent any sort of damage. But, without an automated system, swift evaluation of tens of thousands customer feedback is impossible.

Advances in natural language processing (NLP) algorithms have enabled the development of automated CRM systems. Companies are using these algorithms to determine their marketing strategies by observing their customers opinion about their products [3], [4]. However, sentiment analysis has not been widely investigated for agglutinative languages, such as Turkish. In [5] sentiment polarities of Turkish written movie critics data set were analyzed using an N-gram language model. Kaya et. al. [6] applied a maximum entropy and N-gram language model to classify sentiments of political news from several Turkish news sites . In some studies, a lexicon based approach is applied to conduct sentiment analysis on a movie critics data set [7] [8]. Lately, algorithmic innovations on NLP has enabled the emergence of various word embedding algorithms, among which the most popular is Word2Vec [9]. To the best of our knowledge, as of yet the performance of Word2vec for sentiment analysis in Turkish written text has not yet been investigated.

This paper proposes the use of unsupervised word/document embedding methods for sentiment analysis of Turkish written customer reviews and their further categorization to one of twelve classes. Word2vec is used to capture semantics of words from unlabeled large corpora of customer reviews. After which, a classifier is trained upon word embeddings of labeled training samples for the binary sentiment analysis task, where each word is proportioned by their tf-idf values, then averaged in order to have an averaged review vector for each customer review. Then, a document embedding algorithm, Doc2Vec [10] is trained on negatively classified customer reviews to extract document embeddings for customer reviews. Lastly, a classifier is trained upon document embeddings for the discrimination of the 12 pre-determined categories. Results of both sentiment analysis and categorization are compared with a baseline model: an xgboost classifier trained upon a bag of words vectors. The justification of not choosing a deep neural network approach is that we do not have enough labeled samples to feed the deep neural network. Neural networks are required huge amounts of data in order to yield a good generalization [11]. Also, the usage of transfer learning [12] is not possible since there are no models that have been trained

with a Turkish written data set.

The paper is structured as follows: In Section II, details about the evaluated data set is presented. In section III details for the proposed method is given. Finally, in Section IV, results of both sentiment analysis and categorization are shared and discussed.

II. Data Set

The data set evaluated in this work was collected by a private airline company. The company directly asked its customers about their opinions of their journey in overall, from airport to final destination. The data set contains a total of 14000 customer reviews ($\approx$ 532000 words after pre-processing) written in Turkish, where 1070 of them are labeled. The labeled part of the data set consist of labels for both sentiments and specific categories of reviews. The number of reviews and their average length for each sentiment are shown in Table I. There are 12 specific categories namely, flight crew, customer loyalty program, pantry, overall satisfaction, seat, baggage, boarding, in-flight entertainment (IFE), catering, time performance, lounge, check-in. Positive reviews are assigned to only one category: overall satisfaction. While positive reviews are made up of short sentences in general, negative reviews are complex and long. In addition, the distribution of negative reviews by category is disproportionate, as can be seen from Table II. Examples of customer reviews are given in Table III, translated to English for the benefit of readers.

III. Methodology

The methodology used in this paper is summarized in Figure 1. First, the customer reviews are pre-processed in order to reduce the data complexity for word embedding methods. Then, word embeddings are calculated by using Word2Vec [9] [13] on unlabeled corpus. Furthermore, the word vectors are proportioned by their tf-idf values and then averaged in order to have a single review vector for each review. Then, an extreme gradient boosting (Xgboost) [14] classifier is trained on [15] review vectors of labeled customer reviews for the task of binary sentiment analysis, i.e. classification of positive and negative sentiments. The trained model is then used to classify all the CRM data in order to subtract positive sentiments from the data set. After the sentiment analysis, further categorization of negative sentiments are analyzed. Compared to the binary sentiment task, categorization of reviews is more challenging as the class complexity is higher as well as the labeled samples are being imbalanced. For the categorization of the negative comments, a paragraph embedding method Doc2Vec is employed [10], but only on the reviews which are indexed as negative by the first classifier. Since the labeled data set is imbalanced, the Synthetic Minority Oversampling Technique (SMOTE) [16] is applied to the negative reviews to solve the imbalanced learning problem. Then another Xgboost classifier is trained on document vectors for the categorization task. 10 fold cross validation is applied in training of models. Aforementioned steps are explained in detail in the following subsections.

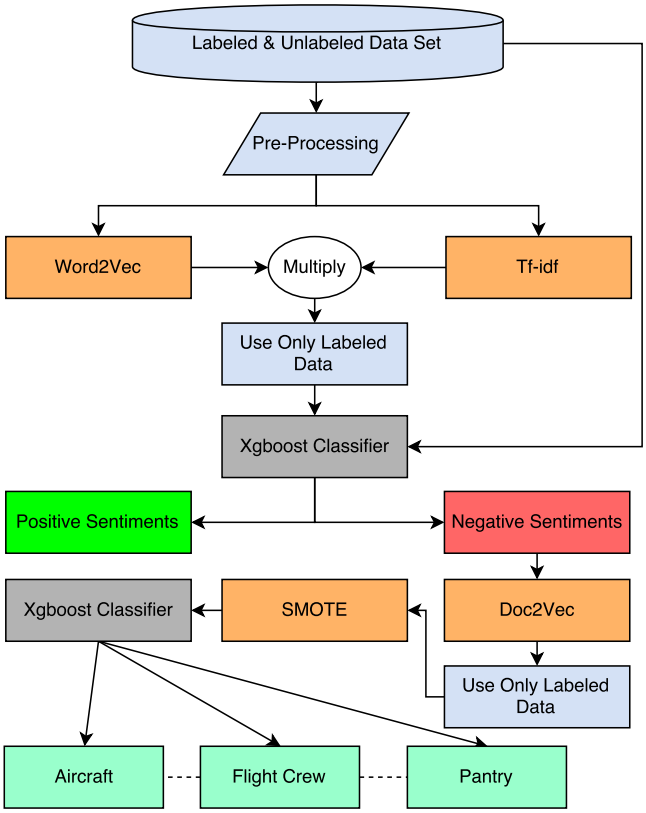

Fig. 1. Flow diagram of the proposed algorithm

A. *Pre Processing*

To obtain proper word embeddings, a pre-processing stage is essential. Thus, the data set cleansed from numbers, punctuations and stop words. Lemmatization and tokenization is also applied.

Tokenization) Tokenization is an operation which splits a given sentence into individual words. However, Turkish contains several non-ascii letters, namely, 'ı', 'ç', 'ğ', 'ş', 'ö', 'ü', which makes this problematic for standard tokenizers. In this study, Zemberek, an open source tokenization and deascification library specifically developed for Turkish language is used [17].

Elimination of Stopwords, Punctuations and Numbers) Stop words are referring to the frequently used words. In this work, 165 words such as 'fakat, de, da, ama, en, ki, ve' are considered stop words of which holds conjunctions and pronouns. These are removed from the data set but adjectives, such as good, nice etc. are kept as they are related to the subject of interest. Punctuations are fairly irrelevant in the data set. For example, exclamation mark is both used in negative and positive sentiments nearly the same amount. Therefore, all punctuations are discarded. Numbers also contain very little or no information considering the objective of this work.

Lemmatization) Lemmatization is an important operation in order to reduce the word complexity. As mentioned, Turkish is a agglutinative language, which makes stemming/lemmatizing

TABLE I
DISTRIBUTION OF LABELED DATA ACCORDING TO THE POSITIVE AND NEGATIVE SENTIMENTS AND THEIR AVERAGE LENGTH

Sentiments	Number of Reviews	Average Length After Pre Processing (in terms of words)
Positive	406	21.7
Negative	664	48.9

TABLE II
NUMBER OF LABELED REVIEWS FOR EACH NEGATIVE SENTIMENT CATEGORY

Class	**Flight Crew**	**Customer Loyalty Program**	**Pantry**	**Overall Satisfaction**	**Seat**	**Baggage**	**Boarding**	**Ife**	**Catering**	**Time Performance**	**Lounge**	**Check-in**
Sample Size	120	112	80	80	47	42	39	32	35	29	26	23

TABLE III
SOME EXAMPLE REVIEWS (TRANSLATED TO ENGLISH)

Feedback	**Category**	**Sentiment**
Everything was great, thank you. Keep on going!	Overall Satisfaction	Positive
The call centre I contacted about my luggage delay were extremely unhelpful. They misinformed me about where to file my complaint, took 17 days to reply to my email and most importantly, they did not solve my problem.	Baggage	Negative

difficult. In this work, an open source lemmatization library turkish-lemmatizer, which is specifically designed for Turkish language, is employed [18]. The library uses longest matched stemming algorithm for lemmatization .

B. Feature Extraction

In this paper, unsupervised word/document embedding methods are employed, such as Word2Vec and Doc2Vec, for feature extraction. The word vectors created by Word2vec are averaged by their TF-IDF values to have a 'review vector' for each customer review. Also, for baseline model the bag of words technique is used for feature extraction.

Word2Vec) Word2Vec is a word embedding method that has been shown [9] to be useful as it preserves the semantics of words in unsupervised manner. Word2Vec is a shallow neural network in general, which has one input, one hidden and one output layers. There are two Word2Vec models available; Continuous Bag of Words (CBOW) and Skip-Gram. In CBOW, model predicting a word from its surrounding words, thus the order of words are ignored. In Skip-Gram, which is the opposite of CBOW, the model is predicting the context from the given word. In this paper, the Skip-Gram approach is used as it takes into account the order of the words. For detailed mathematical explanation of Skip-Gram approach, authors recommend reading [19] and references therein. A simplified explanation can be given as follows: Let w denote the corpus of words and let c be the context of words for a given data set D. The skip-gram model is trying to maximize the conditional probability $p(c|w)$ by optimizing its parameters θ. Thus the objective function can be given given as:

$$\arg\max_{\theta} \prod_{(w,c)\in D} p(c|w;\theta) \tag{1}$$

Each word in corpus needs to be encoded into one hot vectors in order to be used in the model. Let v_c and v_w be the encoded versions of c and w respectively and let C represent the whole context. To maximize Equation 1, the softmax function is employed:

$$p(c|w;\theta) = \frac{e^{v_c.v_w}}{\sum_{(c'\in C)} e^{v_{c'}.v_w}} \tag{2}$$

Nominator of Equation 2 is the dot product between an encoded word vector v_w and its context v_c. Intuitively, the related words, i.e. the words in the same context, should yield a higher dot product value compared to the unrelated words. On denominator, c' refers to all contexts for a given corpus. It is computationally very expensive to calculate all word pairs, therefore an approximation is needed. In order to prevent this bottleneck, the authors of Word2Vec developed a method called *negative-sampling*. Negative sampling states that, if some unrelated w, c pairs are added to the network by creating D' from random w, c pairs, the network learns a unique representation for each word.

Gensim has a popular Word2Vec implementation of which we have used in this work [20]. Word2Vec implementation of Gensim requires some hyperparameters to be tuned. In this work, hyperparameters are determined empirically where vector dimensionality for each word is selected as 200, context size of 10 and downsampling factor of 10^{-3} is used. In order to observe the quality of the word embeddings, we have investigated some of the key words. For example, the word 'koltuk' ('seat' in English) is most similar (yields a high dot product value) to the words; dar(narrow), geniş(wide), boy(size/length). The word eğlence (entertainment) is most similar to the words altyazı (subtitle), Türkçe (Turkish) and sistem (system).

Bag of Words) Bag of words algorithm is based on creating a document vector by word counts [21]. The algorithm creates a histogram-like document vector based on word count i.e. by counting each word that appears more than the given threshold for a given document. In this work, threshold value is selected as 4000, which indicates that we use the most frequent 4000 words. The value of threshold is determined empirically.

Tf-idf) TF-IDF is the abbreviation of the term frequency inverse document frequency. Term frequency measures the frequency of terms occurring in the document. Inverse document frequency measures the importance of words. The IDF coefficients are often very useful for weighting frequent words. Because, some words might occur more than others which might impact the vectorization quality of customer reviews. Thus, instead of directly averaging word embeddings of each word in a customer review, it is beneficial to calculate review vectors by proportioning each word embedding with their idf value.

Doc2Vec) Doc2Vec is an unsupervised learning algorithm, which aims to find the embeddings of documents. The Doc2Vec algorithm, is implemented by adding a paragraph vector to the aforementioned Word2Vec algorithm. Similar to Word2Vec, there are two Doc2Vec models, namely, Distributed Memory (similar to CBOW) model and Distributed Bag of Words (similar to Skip-Gram) model. While the latter ignores word ordering, the former keeps it by concatenating the paragraph vector and word vectors in order to predict the next word in the given context. Doc2Vec algorithm has two advantages; first, it preserves word order and second, it is an unsupervised learning algorithm. Keeping the word order is seen to be essential in categorization task as it is much more complicated compared to the binary sentiment analysis task. Also, being an unsupervised learning algorithm makes Doc2Vec suitable for this task as we have a large corpus of unlabeled comments, where Doc2Vec can learn semantics of customer comments without in need of the labels. Gensim also has an implementation of Doc2Vec of which we have used in this work where document dimensionality is selected as 200, context size of 10 is used and downsampling factor of 10^{-3} is used.

C. *Post Processing*

As mentioned previously, the review categories are somewhat imbalanced. In order to prevent imbalanced learning SMOTE is employed. SMOTE creates synthetic samples in the local neighbors of features by subtracting the feature vector from its nearest neighbor then multiplies the result by a random number between 0 and 1 and adds it to the feature vector. In order to prevent overfitting, SMOTE is applied only to the training data.

D. *Classification Model*

The extreme gradient boosting (Xgboost) algorithm is selected for the classification task. Xgboost is a supervised tree boosting algorithm which combines many weak learners to produce a strong learner. For the given training samples x_i and their labels y_i, Xgboost algorithm uses K weak learners to predict the output $\bar{y_i}$:

$$\bar{y_i} = \sum_{k=1}^{K} f_k(x_i) \tag{3}$$

Here f_k denotes a tree structure which contains a continuous score w_i on its i_{th} leaf. The score of each tree is calculated by minimizing the following objective function

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \bar{y_i}^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{4}$$

where l denotes a convex loss function which can be differentiated in order to measure the difference between y_i and $\bar{y_i}^{(t)}$. Here $\bar{y_i}^{(t)}$ denotes the prediction of the i_{th} sample at t_{th} iteration. Ω is the regularization term where $\Omega(f_t) = \frac{1}{2}\lambda||w||^2$. The regularization term prevents leaf scores to have large values. $f_t(x_i))$'s of which decreases the Equation 4 are greedily added to the tree to obtain the final classification tree. Detailed explanation of Xgboost algorithm is given in [14].

IV. Results

Even though the labeled data set is not very large, the usage of unsupervised techniques such as Word2Vec and Doc2Vec made it possible for us to utilize the large unlabeled corpus that we have. The confusion matrices of both the categorization task and the sentiment analysis are given on Table-IV and Table-V where classification accuracies are reported as 71.16% and 92.5% respectively. For both tasks, our approach surpasses the baseline model by a great margin, where we have obtained 75.7% accuracy for sentiment analysis and 52.1% accuracy for categorization by utilizing a bag of words approach. It is important to note that, the baseline method is implemented in a non-hierarchical way as we considered sentiment analysis and categorization separately.

By analyzing the results of the sentiment analysis, we report that the confusion between sentiments is caused by the reviews that are comprised of 'neutral' emotion of which we have not investigated in this work. Furthermore, most of the confusions between classes in the categorization task are dependent upon two main reasons: First and foremost, we assumed that a customer feedback is only related to a certain category, however some reviews contain multiple categories. For example, feedback related to the lounge are confused with the customer loyalty program, which is intuitive as in most of the feedback many customers have mentioned that they need to be given better rights at lounge when utilizing the customer loyalty program. Same phenomena applies for some other classes as well. Secondly, the classes with high error are seen to have the classes that have small number of training samples where SMOTE is failed to generate proper samples. Authors conclude that, instead of multi-class classification the categorization task can be thought as a multi-label classification, where a single feedback can be comprised of multiple labels.

TABLE IV
CONFUSION MATRIX FOR THE CATEGORIZATION OF NEGATIVE REVIEWS

	Overall Satisfaction	Boarding	Check In	Pantry	Seat	Baggage	Flight Crew	Ife	Catering	Time Performance	Lounge	Customer Loyalty Program
Overall Satisfaction	**0.75**	**0.09**	0	0.03	0.02	0.05	0	0	0.04	0.01	0	0.01
Boarding	**0.06**	**0.67**	0.01	0	**0.1**	0.01	**0.06**	**0.07**	0.01	0	0.01	0
Check In	0.03	0.02	**0.74**	0	0.04	0	0.03	**0.06**	0	0	0.05	0.03
Pantry	0	0.03	0	**0.88**	0.01	0	0.03	0	0	0.05	0	0
Seat	0.01	0.04	**0.06**	0	**0.53**	**0.06**	0.02	0.02	**0.09**	0.04	**0.06**	**0.07**
Baggage	0	0	0.03	0.01	0.05	**0.78**	0	0.01	0.03	0	**0.09**	0
Flight Crew	0	0.03	0	0	0.03	0	**0.82**	0	0.01	**0.06**	0.05	0
Ife	0.01	0.01	0	0.01	0.01	0.01	0.01	**0.73**	0.04	**0.1**	**0.07**	0
Catering	0.01	0	0	0	0.01	0	0.01	0.03	**0.83**	0.04	0.03	0.04
Time Performance	0.05	0	0.05	0	**0.07**	0.03	0.04	**0.12**	0.03	**0.49**	**0.12**	0
Lounge	0.02	0	0.01	0.01	**0.07**	**0.06**	0.04	**0.07**	**0.06**	0.04	**0.52**	**0.1**
Customer Loyalty Program	0	0.01	0.01	0	0.03	0.01	0.01	0.03	0.05	0.03	0.02	**0.8**

TABLE V
CONFUSION MATRIX FOR SENTIMENT ANALYSIS

	Positive	Negative
Positive	0.885	0.115
Negative	0.035	0.965

V. ACKNOWLEDGEMENT

Thanks to STM Defense Technologies Engineering and Trade Inc. for supporting this study. STM provides system engineering, technical support, project management, technology transfer and logistics support services for TAF (Turkish Armed Forces) and SSM (Undersecretariat for Defense Industries).

REFERENCES

[1] Y.-C. Ku, C.-P. Wei, and H.-W. Hsiao, "To whom should i listen? finding reputable reviewers in opinion-sharing communities," *Decision Support Systems*, vol. 53, no. 3, pp. 534–542, 2012.

[2] L. D. Peters, A. D. Pressey, and P. Greenberg, "The impact of crm 2.0 on customer insight," *Journal of Business & Industrial Marketing*, vol. 25, no. 6, pp. 410–419, 2010.

[3] T. Miyoshi and Y. Nakagami, "Sentiment classification of customer reviews on electric products," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*. IEEE, 2007, pp. 2028–2033.

[4] P. Gunarathne, H. Rui, and A. Seidmann, "Customer service on social media: The effect of customer popularity and sentiment on airline response," in *System Sciences (HICSS), 2015 48th Hawaii International Conference on*. IEEE, 2015, pp. 3288–3297.

[5] U. Erogul, "Sentiment analysis in turkish," *Middle East Technical University, Ms Thesis, Computer Engineering*, 2009.

[6] M. Kaya, G. Fidan, and I. H. Toroslu, "Sentiment analysis of turkish political news," in *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*. IEEE Computer Society, 2012, pp. 174–180.

[7] A. G. Vural, B. B. Cambazoglu, P. Senkul, and Z. O. Tokgoz, "A framework for sentiment analysis in turkish: Application to polarity detection of movie reviews in turkish," in *Computer and Information Sciences III*. Springer, 2013, pp. 437–445.

[8] C. Türkmenoglu and A. C. Tantug, "Sentiment analysis in turkish media," in *Proceedings of Workshop on Issues of Sentiment Discovery and Opinion Mining, International Conference on Machine Learning (ICML), Beijing, China*, 2014.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[10] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1188–1196.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[12] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[14] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794.

[15] M. U. Çakir and S. Güldamlasioglu, "Text mining analysis in turkish language using big data tools," in *Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual*, vol. 1. IEEE, 2016, pp. 614–618.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[17] A. A. Akın and M. D. Akın, "Zemberek, an open source nlp framework for turkic languages," *Structure*, vol. 10, pp. 1–5, 2007.

[18] Baturman, "Lemmatization in turkish language," https://github.com/baturman/turkish-lemmatizer/wiki/Lemmatization-in-Turkish-Language, 2013.

[19] Y. Goldberg and O. Levy, "word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722*, 2014.

[20] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[21] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.