Using Word Embeddings in Turkish Part of Speech Tagging

Şevket Can, Bahar Karaoğlan, Tarık Kışla, and Senem Kumova Metin

***Abstract*—The close relation between the stem (relatively the word meaning) and part of speech tag of the word turns part of speech tagging as an important preprocessing task in natural language processing and understanding problem. For example, if the Turkish word "gelecek" is labeled as noun, the word stem is to be "gelecek" meaning future. If it is labeled as verb, the stem is "gel" and in English it means, "come". In many languages including Turkish, part of speech tagging problem is generally solved by rule based approaches. In this paper, a setup where the neural network architecture SENNA together with word embeddings is employed. The combination of Wikipedia 2016 and METU corpora is utilized in training of word embeddings; PARDER is used in part of speech training and testing. The word embeddings that are obtained by different methods and different vector sizes are evaluated intrinsically considering analogic and semantic similarity distances; and assessed extrinsically based on the performance on part of speech tagging task.**

***Index Terms*—Part of speech tagging, word embedding, SENNA, deep learn.**

I. Introduction

In natural language processing, part of speech tagging (POS) is defined as the mapping of words to their corresponding part of speech tags in a text. The POS tagging is important in many different fields such as information retrieval, natural language generation, and automatic translation. Though there exists different categorization in different resources, part of speech tags may be classified in eight main categories in Turkish: noun, verb, adjective, adverb, pronoun, conjunction, question and preposition. In Table I, regarding tags and some Turkish examples are given.

The main difficulty in POS tagging is that a single word may have a different part of speech tag in different sentences based on the contexts. This is why POS tag of a word must be determined according to the context. Following, three sentences are given as examples, where the word "gelecek" should be tagged as adjective, noun and verb with respective to three different meanings (Eng. "future", "next" and "will come").

Manuscript received November 9, 2019; revised October 11, 2020.

Şevket Can is with the International Computer Institute, Ege University, Izmir, Turkey (e-mail: sevketcann@ gmail.com).

Bahar Karaoğlan is with the International Computer Institute, Ege University, Izmir, Turkey (e-mail: bahar.karaoglan@ege.edu.tr).

Tarık Kışla is with the Department of Computer Education and Instructional Technologies, Ege University, Izmir, Turkey (e-mail: tarik.kisla@ege.edu.tr).

Senem Kumova Metin is with the Department of Software Engineering, İzmir University of Economics, İzmir, Turkey (e-mail: senem.kumova@ieu.edu.tr).

TABLE I: The POS Tags and Turkish Examples

TAG	MEANING	TURKISH EXAMPLES
ADJ	ADJECTIVE	yeni (new), çirkin (ugly), yüksek(high), büyük (big), yerel (local)
ADV	ADVERB	really, already, still, early, now
CONJ	CONJUNCTION	ve (and), veya (or), ama (but), eğer (if), iken (while), rağmen (although)
NOUN	NOUN	sene (year), ev (home), zaman (time), kalem (pen), masa (table)
PRON	PRONOUN	o (he, she, it), onlar (they), ben (i), sen (you) ,biz(us), bu (this)
VERB	VERB	sor (ask), yaz (write), koş (run), ye (eat), iç (drink), cevapla (answer)
QUES	QUESTION	kim (who), neden (why), ne (what), niye (why), kimin (whose)
PREP	PREPOSITION	içinde (in), içerisinde (in), üstünde (on), dışında (out of), altında (below)

Example 1. *Gelecek* yıl ekonomi açısından zor geçecektir. (*Next* year will be difficult for the economy)

Example 2. *Gelecek* planlanırken birçok parametre göz önünde bulundurulmalıdır.
(Many parameters should be considered when planning the *future*.)

Example 3. Komşularımız bu akşam bizi ziyarete etmeye *gelecek.*
(Our neighbors *will come* to visit us this evening.)

When the POS tagging studies in the field are examined, it is observed that POS tagging methods are gathered around two main groups: rule-based and statistical approaches. In rule-based approaches, typically contextual information is employed to assign tags to unknown or ambiguous words. Simply, analyzing the linguistic features of the word, its preceding word, its following word, and other aspects disambiguation is performed. For example, in Turkish, if the preceding word is an adjective, then the word in question must be a noun or an adjective. And in rule-based methods, this information must be coded in the form of rules. On the other hand, the part of speech tagging studies based on statistical methods commonly utilize corpora in order to obtain required statistical information. The simplest statistical POS taggers label the words based solely on the probability that a word occurs with a particular tag. In other words, the tag encountered most frequently in the training set with the word is the one assigned to an ambiguous instance of that word. An alternative to this approach is to calculate the probability of a given sequence of tags occurring. This is sometimes referred to as the *n-gram* approach. Since, in *n-gram* approach, the tags of words in a sequence of n words are employed, it is accepted to consider the context while POS tagging. In literature, there also exist methods (e.g. hidden Markov model) that consider both the context and

doi: 10.18178/ijmlc.2021.11.5.1063

word frequencies.

In POS tagging, there exist multiple factors that have influence on the performance of tagger. One of the main factors is the corpus that is utilized in experiments while modeling the tagger language. The corpus size, reliability of labeled data and variety of the corpus has influence on POS tagger performance. For example, Brown corpus was one of the firstly used corpus for POS tagging studies in English. The second factor is the set of the preprocessing tasks such as tokenization. And the last but not the least one is the language. It is known that for different languages the POS tagging approaches may perform differently. Though the performances of POS tagging studies that employ statistical or rule-based methods reach to acceptable values (96%-98%) in English [1]-[8], the performances drop to 80%-92% levels in Turkish studies [9]-[14]. This is due to the agglutinative structure of Turkish and/or the theoretical infinite size of vocabulary in Turkish. In Table II, an example set of POS taggers and the methods that are employed are provided.

TABLE II: THE POS TAGGER-EXAMPLES

TAGGER	METHOD	TAGGER	METHOD
TnT [15]	Hidden Markov model	Stanford Tagger 2.0 [23]	Maximum entropy cyclic dependency network
Melt [16]	Maximum entropy Markov model with external lexical information	SCCN [24]	Semi-supervised condensed nearest neighbor
GENiA Tagger[17]	Maximum entropy cyclic dependency network	CharWNN [25]	MLP with neural character embeddings
Averaged Perceptron [18]	Averaged perceptron	structReg [26]	CRF with structure regularization
Maxent easiest-first [19]	Maximum entropy bidirectional easiest-first inference	BI-LSTM-CRF [27]	Bidirectional LSTM-CRF
LAPOS [20]	Perceptron based training with lookahead	NLP4J [28]	Dynamic feature induction
Flair [21]	Bidirectional LSTM-CRF with contextual string embeddings	SVMTool [29]	SVM-based tagger and tagger generator
Morče/Compost [22]	Averaged perceptron	LTAG-SPINAL [30]	Bidirectional perceptron learning

We propose the use of word embeddings with deep learning methods in order to identify POS tags in Turkish. The word embedding is simply a type of word representation where the text is turned to numbers allowing words with similar meaning to be understood by machine learning algorithms. It is also called as distributed semantic model or distributed represented or semantic vector space or vector space model. In word embedding approach it is accepted that the words convey their meaning with the words occurring in the same context. As a result, fruits like apple, orange should be placed close whereas sports will be far away from these words. In a broader sense, word embedding will create the vector of fruits, which will be placed far away from vector representation of sports. This enables to run simple mathematical operations to detect the semantic relations between words. As in a typical example, it is possible to obtain the embedding of *king* by subtracting the embedding of *woman* from the embedding of *queen.*

In this study, the experiments are performed on SENNA neural network structure developed by Collobert *et al.* [31]. Firstly, the performance of the approach is measured utilizing English Brown corpus to assure the correct employment. Then, the experiments are repeated using Turkish corpus.

In following sections, the method will be presented, experimental results will be given and the paper will be concluded respectively.

II. THE PROPOSED METHOD: WORD EMBEDDINGS IN POS TAGGING

In this study, Turkish word embeddings are generated by word2vec method and the embeddings are given as input to the SENNA tool to mark the part of speech labels of the regarding word.

SENNA (Semantic/Syntactic Extraction using a Neural Network Architecture) tool built by Collobert *et al*. [31] is an architecture that provides machine learning by a neural network. The tool may be used for several tasks (e.g. semantic role labeling, entity recognition) in natural language processing field. The main goal in SENNA is enabling several tasks omitting feature engineering and learning the semantic relations between words in text based on the occurrence frequencies. SENNA is proposed in two different set-ups to be used for different tasks. These set-ups are similar in terms of neural network structure. The difference between them is the approach to generate the required input to the network. These approaches are

i. Window-based: The approach requires determining the neighboring words to the target word and employing their word embeddings. The approach is commonly used for natural language processing problems such as named entity recognition (NER) and POS tagging where the target word is related to the context words.

ii. Sentence-based: In this approach, all the words residing in the same sentence with the target word are considered. It is required to obtain word embeddings of all words to generate the sentence embedding and give sentence embedding as an input to the architecture. It is commonly used in problems such as semantic role labeling where the solution is hidden in the sentence structure.

The window-based approach is employed in our study assuming that each word is related to the neighboring words in a given window. In Fig. 1, window-based approach is exemplified for the sentence "Ayşe okula geç geldi" (Ayşe came to the school late). In this example, target word is "geç" (late); window size is set to 2; "Ayşe", "okula" (to the school) and "geldi" (came) are the context words of the regarding target word.

The tasks that are followed to build up the set-up in Fig. 1 are:

i. The neighboring words of the target word in the given window size=2 are determined as context words and accepted as inputs to the system.
ii. Word embeddings of the contexts words are retrieved from the word embeddings data set.
iii. A merged matrix is built by appending the word embeddings.

iv. The matrix is transformed to a linear data structure by affine transformation.
v. Tangent activation function is applied on the matrix in the hidden layer of the neural network.
vi. The probability values of possible POS tags for each target word are determined by softmax classifier using the transformed matrix.
vii. Finally, the target word is marked with the tag that holds the highest probability value. Based on the probability values, target word "geç" is labeled as "adverb" in given example.

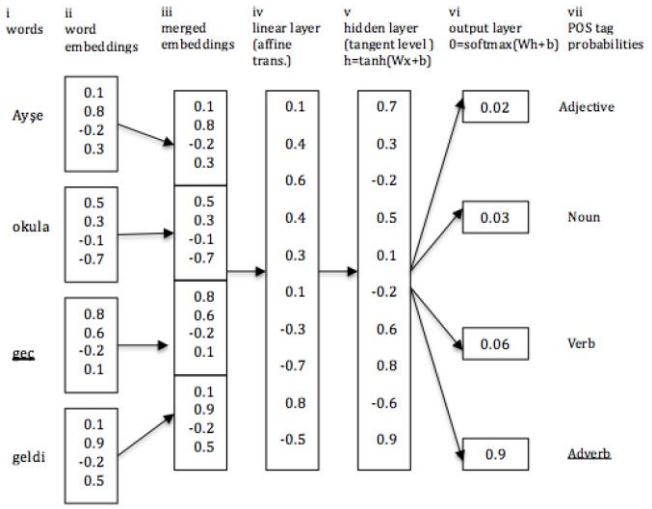


Fig. 1. Window based architecture – example sentence.

A similar procedure may be applied for the sentence-based approach. The difference between the two approaches is that in the sentence-based approach all the words that reside in the same sentence with the target word is considered as its neighboring words and their word embeddings are given as input to the system. In our experiments, we did not run tests by sentence-based approach.

III. Experimental Results

The performance of the proposed approach is firstly in measured on English Brown corpus. Brown corpus in NLTK library is utilized in building training (50545 sentences), validation (2505 sentences) and testing (4134 sentences) data sets. The word embeddings pre-trained by GloVe [32] and EDBSG (Extended Dependency Based Skip-Gram) [33] methods are used.

Skip-gram model predicts surrounding context words given a target word. In Fig. 2, the architecture of the basic skip gram model is depicted. Here, $w(t)$ is the target word and there exists one hidden layer which performs the dot product between the weight matrix and the input vector of $w(t)$. No activation function is used in the hidden layer (depicted as projection layer) and the result of the dot product at the hidden layer is passed to the output layer. Output layer computes the dot product between the output vector of the hidden layer and the weight matrix of the output layer. Then the *softmax* activation function is applied to compute the probability of words appearing to be in the context of $w(t)$ at given context location. As the number of words to be predicted increases, the problem gets more complex.

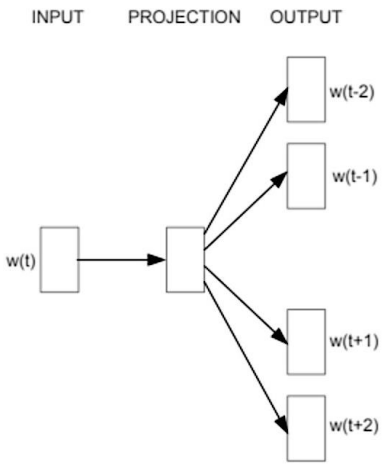


Fig. 2. The basic skip-gram model architecture (Source: https://arxiv.org/pdf/1301.3781.pdf Mikolov *el al*).

GloVe (Global Vectors for Word Representation) [32] is developed by Stanford University to generate vector representations for words. The aim of GloVe is to produce word vectors that find the "meaning in vector space" by using statistics of global count. Distinctly from continuous bag of words or skip gram models, GloVe learns based upon a co-occurrence matrix and trains vectors thus their differences estimate co-occurrence ratios. In GloVe model, global matrix factorization and local context window methods are employed. Here, local context window methods are well-known continuous bag of words and skip-gram methods. The global matrix factorization is used to reduce large term frequency matrices in latent semantic analysis. And also, this method is used in GloVe to include global frequency information in order to build up word vectors. In GloVe model, instead of co-occurrence probabilities the ratio of co-occurrence probabilities is used.

In this experiment, vector size of word embeddings (D) is set to 300 and window size (W) is 5. Table III gives the performance results for English corpus where accuracy values on test (Test_Accuracy) and validation (Val_Accuracy) sets are obtained by running the system ten times (run=10). The term accuracy refers to a statistical measure that presents the ratio of correctly classified samples. It is formulated as below:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

where TP refers to true positives, TN is true negatives, FP is false positives and FN is false negatives.

TABLE III: The Performance Results in English Corpus

Parameters						
W	D	ES	SPB	Run	Val Accuracy	Test Accuracy
10	300	~1.50%	5	10	93.22	93.19
5	300	~2.45%	5	10	95.97	95.98

In the Table III, SPB is the window size used in SENNA tool to label part of speech tags. ES represents proportion of the words that reside in testing set but do not have a valid word embedding.

As given in Table III, the accuracy values for testing set reach to 93.19% and 95.98% by GloVe and EDBSG

embeddings, respectively. The performance results for English corpus are similar to previous studies on POS tagging showing that the proposed set-up is proper to be used in marking part of speech tags.

TABLE IV: THE STATISTICS ON VO CORPUS

CORPUS	WORD SIZE	VOCABULARY SIZE	NUMBER OF SENTENCES
WIKIPEDIA2016	45983505	1220305	~3894241
METU CORPUS	2065079	192998	~151416
VO CORPUS	48048584	1395047	~4045657

A similar experiment is performed on Turkish corpus as the second step of the study. Wikipedia (March 2016-https://dumps.wikimedia.org) articles (Wikipedia2016) and METU corpus [34] are merged to build Turkish corpus (VO). Table IV depicts some statistics on VO, METU and Wikipedia2016 corpora.

Turkish word embeddings are obtained applying word2vec skip-gram method on VO corpus by Gensim tool [35] using surface forms of words. In order to decrease the number of words with missing embeddings, punctuation marks are removed from VO corpus, all numerical entities are labeled as NUM and the words that occur in corpus less than two times are not included in training. Training is repeated for two different window (W=2 and W=5) and vector (word embedding) (D=100 and D=200) sizes. For example, (W, 2, 100) represents the setting where the embedding vector size=100 and window size=2.

TABLE V: THE SIMILAR WORDS TO "TÜRKIYE", "APPLE" AND "AĞUSTOS" WORDS

	TÜRKIYE	**APPLE**	**AĞUSTOS**
Corpus:VO	Kktc (Turkish Republic of Northern Cyprus)	google	aralık (december)
W = 2,	Sscb (USSR-Union of Soviet Socialist Republics)	iphone	şubat (february)
D = 100	Abd (USA)	ios	mart (march)
	İsviçre (Switzerland)	ipod	mayıs (may)
	Trt (abbreviation of Turkish Radio and Television Association)	app	ocak (january)
Corpus:VO	Kktc (Turkish Republic of Northern Cyprus)	google	ocak (january)
W = 2,	Sscb (USSR-Union of Soviet Socialist Republics)	ios	aralık (december)
D = 200	turkiye	android	şubat (february)
	İngiltere (England)	nokia	mart (march)
	Tbmm (GNAT- Grand National Assembly of Turkey)	microsoft	mayıs (may)
Corpus:VO	Kktc (Turkish Republic of Northern Cyprus)	google	şubat (february)
W = 5,	Fenerbah œ (A famous sports club in Turkey)	ios	ocak (january)
D = 100	Trt (abbreviation of Turkish Radio and Television Association)	ipod	aralık (december)
	Tbmm (GNAT- Grand National Assembly of Turkey)	android	mayıs (may)
	Kayseri (A city in Turkey)	iphone	mart (march)
Corpus:VO	Kktc (Turkish Republic of Northern Cyprus)	google	şubat (february)
W = 5,	Kayseri (A city in Turkey)	ios	mayıs (may)
D = 200	Tbmm (GNAT- Grand National Assembly of Turkey)	zune	aralık (december)
	İsviçre (Switzerland)	iphone	ocak (january)
	Sscb (USSR-Union of Soviet Socialist Republics)	ipod	mart (march)

The semantic similarities of the embedding vectors for target words "Türkiye" (Turkey), "Apple" and "ağustos" (august) are given in Table III as examples. Cosine similarity is used in measuring semantic similarity between given couple of embedding vectors. In Table V, the first row includes the target words, each word in the regarding column shows the most similar words to the target word. For example, the target word "Apple" is similar to "google", "iphone", "ios","ipod" and "app" in order when W=2 and D=100. The sorted list of similar words to a group of target words (such as the ones in Table III), showed that word embeddings are quite successful to represent the words in Turkish.

Following the retrieval of word embeddings, Turkish part of speech training and testing tasks are performed on PARDER [36] Turkish corpus. The corpus is split into three parts as training set of 12397 sentences, testing set of 1535 sentences and validation set of 1104 sentences. The word in PARDER corpus is labeled with 17 different POS tags (adjective, adverb, conjuction, determinant, duplication, interjunction, Ndet, Ndot, Ntime, Nnum, noun, number, post-pronoun, pronoun, punctuation, question, verb).

Table VI gives the experimental results obtained from Turkish corpus. In Table VI, W is window size; D is vector size, and ES represents the proportion of the words that do not have valid embeddings. While the experiments are repeated with different W and D values all units referring to numbers are accepted as a single word. The number of iterations in each experiment is set to 10 and POS window size is determined as 3. In the initial experiments, it is observed that most of the words that do not have valid embeddings are punctuation marks. As a result, word embeddings for all punctuation marks are generated and the experiments are repeated. For example, for the punctuation mark ".", a word embedding (vector) is built with the given size. After this correction, it is observed that the proportion of such words are decreased from ES=13.50% to ES=~3.75%. The experimental results before the word embedding correction of punctuation marks (ES=13.50%) are given in first two rows of Table VI. The accuracy values when ES value is lowered to 3.75% are presented in third and fourth rows on Table VI.

Examining Table VI, it is seen that the highest accuracy value (83.09%) is obtained when windows size is set (W) to 5 and vector size D=200.

TABLE VI: THE PERFORMANCE RESULTS IN TURKISH CORPUS

EXPERIMENT NO	PARAMETERS W	D	ES	VAL ACCURACY	TEST ACCURACY
1	5	100	~13.50%	82.81	82.03
	5	200	~13.50%	83.16	83.09
2	5	100	~3.75%	81.87	81.75
	5	200	~3.75%	82.39	82.38

IV. CONCLUSION

In this study, a set-up that uses SENNA tool with word embeddings is proposed to label Turkish words with proper part of speech tags. Though the experimental results showed

that the proposed set-up is quite successful, there exists a room for improvement since the performance values are still lower compared to existing 80%-92% accuracy values in previous Turkish studies. We believe that the performance for Turkish may be improved by increasing the number of samples in training set. In order to test the change in performance as the size of training set is increased, we repeated the tests on different sizes of training set. The tests are performed on English corpus since there exists still not enough data samples in Turkish. In Fig. 3, horizontal axis represents the number sentences in training corpus and vertical axis holds the accuracy values. It may be examined from Fig. 3 is that as the size of the training set is changed from 10.000 to 50.000 samples, the accuracy value continuously increases supporting our claim.

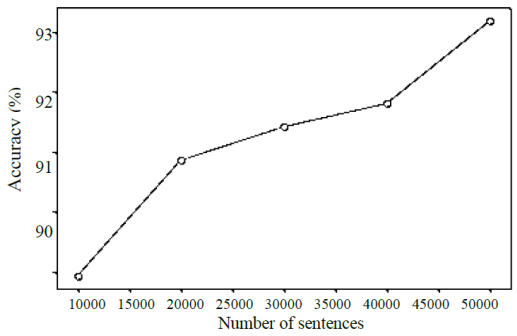


Fig. 3. The impact of the training set size on POS tagging performance (GloVe word embeddings are used in experiment).

As a further study, we plan to increase the training data set size in Turkish, change the number of levels in neural network structure in order to improve POS tagging performance in Turkish.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Şevket Can conducted the experiments; Bahar Karaoğlan, Tarık Kışla and Senem Kumova Metin analyzed the data and wrote the paper; all authors had approved the final version.

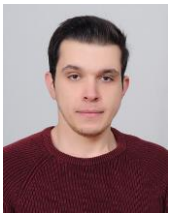
Acknowledgment

This work is carried under the grant of Ege University Scientific Research Projects Committee, Project no: 18-UBE-002, Project Title: Part of Speech Tagging with Deep Learning Methods.

References

[1] E. Brill, "A simple rule-based part-of-speech tagger," in *Proc. ANLP-92, 3rd Conference on Applied Natural Language Processing*, Trento, IT, 1992, pp. 152–155.
[2] E. Brill, "Some advances in transformation-based part of speech tagging," in *Proc. the Twelfth National Conference on artificial Intelligence, AAAI'94*, 1994, pp. 722–727.
[3] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, Cambridge, Mass.: MIT Press, 1999.
[4] D. Jurafsky and J. H. Martin, *Speech and language Processing*, 2nd ed. Prentice Hall, 2008.
[5] A. McCallum, D. Freitag, and F. Pereira, "Maximum entropy Markov models for information extraction and segmentation," in *Proc. 17th International Conference on Machine Learning*, San Francisco, CA: Morgan Kaufmann, 2000, pp. 591–598.
[6] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th International Conf. on Machine Learning*, San Francisco, CA: Morgan Kaufmann, 2001, pp. 282–289.
[7] H. Zang and J. DeNero, "Observational initialization of type-supervised taggers," in *Proc. the 52nd Annual Meeting of the Association for Computational Linguistics*, June 2014, vol. 2, pp. 816-821.
[8] K. Stratos, M. Collins, andD. Hsu, "Unsupervised part-of-speech tagging with anchor hidden Markov models," *Transactions of the Association for Computational Linguistic*s, vol. 4, pp. 245-257, 2016.
[9] K. Oflazer and I. Kuruoz, "Tagging and morphological disambiguation of Turkish text," in *Proc. the 4th Conference on Applied Natural Language Processing*, 1994, pp. 144-149.
[10] D. Z. H. Tur, K. Oflazer, and G. Tur, "Statistical morphological disambiguation for agglutinative languages," *Computers and the Humanities*, vol. 36, pp. 381-410, 2002.
[11] B. T. Dinçer, B. Karaoğlan, and T. Kışla, "A suffix based part-of-speech tagger for Turkish," in *Proc. International Conference on Information Technology: New Generations*, 2008, pp. 680-685.
[12] R. Ehsani, M. E. Alper, G. Eryigit, and E. Adali, "Disambiguating main POS tags for Turkish," in *Proc. the 24th Conference on Computational Linguistics and Speech Processing*, 2012.
[13] B. Can, A. Ustün, and M. Kurfalı, "Turkish POS tagging by reducing sparsity with morpheme tags in small datasets," in *Proc. the 17th International Conference on Intelligent Text Processing and Computational Linguistics*, 2016.
[14] N. Bolucu and B. Can, "Stem-based PoS tagging for agglutinative languages," in *Proc. 25th Signal Processing and Communications Applications Conference*, Turkey, 2017.
[15] T. Brants, "TnT -- a statistical part-of-speech tagger," in *Proc. 6th Applied Natural Language Processing Conference*, 2000.
[16] P. Denis and B. Sagot. "Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort," in *Proc. PACLIC 2009*, 2009
[17] Y. Tsuruoka, Y. Tateishi, K. Jin-Dong, O. Tomoko, J. McNaught, S. Ananiadou, and J. Tsujii, "Developing a robust part-of-speech tagger for biomedical text, advances in informatics," in *Proc. 10th Panhellenic Conference on Informatics*, 2005, pp. 382-392.
[18] M. Collins, "Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms," in *Proc. EMNLP*, 2002.
[19] Y. Tsuruoka and J. Tsujii, "Bidirectional inference with the easiest-first strategy for tagging sequence data," in *Proc. HLT/EMNLP 2005*, pp. 467-474.
[20] Y. Tsuruoka, M. Yusuke, and J. Kazama, "Learning with lookahead: Can history-based models rival globally optimized models?" in *Proc. the Fifteenth Conference on Computational Natural Language Learning*, 2011, pp. 238–246.
[21] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proc. COLING*, 2018
[22] D. J. Spoustová, J. Hajič, J. Raab, and M. Spousta, "Semi-supervised training for the averaged perceptron POS tagger," in *Proc. the 12 EACL*, 2009, pp. 763-771.
[23] C. D. Manning, "Part-of-speech Tagging from 97% to 100%: Is it time for some linguistics?" in *Proc. 12th International Conference on Computational Linguistics and Intelligent Text Processing*, 2011, pp. 171-189.
[24] A. Søgaard, "Semi-supervised condensed nearest neighbor for part-of-speech tagging," in *Proc. The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*. Portland, Oregon, 2011.
[25] C. dos Santos, and Z. Bianca, "Learning character-level representations for part-of-speech tagging," in *Proc. the 31st International Conference on Machine Learning*, 2014.
[26] X. Sun, "Structure regularization for structured prediction," *Neural Information Processing Systems (NIPS)*, pp. 2402-2410, 2014
[27] Z. H. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," *arXiv:1508.01991*, 2015.
[28] D. J. Choi, "Dynamic feature induction: The last gist to the state-of-the-art," in *Proc. NAACL*, 2016.
[29] J. Giménez and L. Márquez, "SVMTool: A general POS tagger generator based on support vector machines," in *Proc. the 4th International Conference on Language Resources and Evaluation*, Lisbon, Portugal, 2004.
[30] L. Shen, G. Satta, and A. Joshi, "Guided learning for bidirectional sequence classification," in *Proc. the 45th Annual Meeting of the Association of Computational Linguistics*, 2007, pp. 760-767.
[31] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, 2011.

[32] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.
[33] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "word2vec-v1," in *Proc. Int. Conf. Learn. Represent*, 2013, pp. 1–12.
[34] D. Zeyrek, Ü. Turan, C. Bozsahin, R. Çakici, A. Sevdik-Çalli, İ. Demirşahin, and H. Ögel, "Annotating subordinators in the Turkish discourse bank.," in *Proc. the Third Linguistic Annotation Workshop*, Association for Computational Linguistics, 2009, pp. 44-47.
[35] R. Rehurek and P. Sojka, "Software framework for topic modeling with large corpora," in *Proc. the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010.
[36] B. Karaoğlan, T. Kışla, S. K. Metin, U. Hürriyetoğlu, and K. Soleymanzadeh, "Using multiple metrics in automatically building Turkish paraphrase corpus," *Research in Computing Science*, pp. 75-83, 2016.



Sevket Can has taken his B.S. degree from the Department of Mathematics and Computer Science, Ege University in 2016, and M.S. degree from International Computer Institute, Ege University in 2019. He is currently working as a scholar in 18-UBE-002 dunded by Ege University Scientific Research Projects Committee.

Bahar Karaoglan has taken her B.S. degree from Electrical and Electronics Engineering, Bogazici University in 1977, M.S. degree from Computer Science, Bogazi ǫ University in 1979, and PhD degree from Computer Engineering, Ege University in 1991. She is a professor in International Computer Institute of Ege University, İzmir, Turkey; Turkish scientific committee head in EU MedNet'U (Mediterenean Network of Universities) project; project leader in several national projects funded by Scientific and Technological Research Council of Turkey (TÜBİTAK), and Ege University Science and Technology Application and Research Center. She is giving information retrieval, multimedia systems, computer architecture, information systems and expert systems courses.

Tarık Kışla has taken his B.S. degree in mathematics from Ege University in 1998, M.S and PhD degrees from International Computer Institute. He is currently working as an associate professor in the Department of Computer and Instructional Technologies, Ege University. He gives lectures on information technologies, algorithms, programming language, databases, web design and networks. He is mainly interested in natural language processing, and instructional technologies.

Senem Kumova Metin has taken her B.S degree from Electrical and Electronics Engineering Department, Ege University in 2001, M.S. and PhD degrees from International Computer Institute, Ege University, Izmir in 2005 and 2011. She is currently working as an associate professor in the Department of Software Engineering, Izmir University of Economics. She is mainly interested in natural language processing and machine learning applications.