Categorization of Turkish News Documents with Morphological Analysis

Burak Kerim Akkuş
Computer Engineering Department
Middle East Technical University
Ankara, Turkey
burakkerim@ceng.metu.edu.tr

Ruket Çakıcı
Computer Engineering Department
Middle East Technical University
Ankara, Turkey
ruken@ceng.metu.edu.tr

Abstract

Morphologically rich languages such as Turkish may benefit from morphological analysis in natural language tasks. In this study, we examine the effects of morphological analysis on text categorization task in Turkish. We use stems and word categories that are extracted with morphological analysis as main features and compare them with fixed length stemmers in a bag of words approach with several learning algorithms. We aim to show the effects of using varying degrees of morphological information.

1 Introduction

The goal of text classification is to find the category or the topic of a text. Text categorization has popular applications in daily life such as email routing, spam detection, language identification, audience detection or genre detection and has major part in information retrieval tasks.

The aim of this study is to explain the impact of morphological analysis and POS tagging on Turkish text classification task. We train various classifiers such as k-Nearest Neighbours (kNN), Naive Bayes (NB) and Support Vector Machines (SVM) for this task. Turkish NLP tasks have been proven to benefit from morphological analysis or segmentation of some sort (Eryiğit et al., 2008; Çetinoğlu and Oflazer, 2006; Çakıcı and Baldridge, 2006). Two different settings are used throughout the paper to represent different degrees of stemming and involvement of morphological information. The first one uses the first n-characters (prefixes) of each word in a bag of words approach. A variety of number of characters are compared from 4 to 7 to find the optimal length for data representation. This acts as the baseline for word segmentation in order to make the limited amount of data less sparse. The second setting involves word stems that are extracted with a morphological analysis followed by disambiguation. The effects of part of speech tagging are also explored. Disambiguated morphological data are used along with the part of speech tags as informative features about the word category.

Extracting an n-character prefix is simple and considerably cheap compared to complex state-of-the-art morphological analysis and disambiguation process. There is a trade-off between quality and expense. Therefore, we may choose to use a cheap approximation instead of a more accurate representation if there is no significant sacrifice in the success of the system. Turkish is an agglutinative language that mostly uses suffixes[1]. Therefore, approximate stems that are extracted with fixed size stemming rarely contain any affixes.

The training data used in this study consist of news articles taken from Milliyet Corpus that contains 80293 news articles published in the newspaper Milliyet (Hakkani-Tür et al., 2000) [2]. The articles we use for training contain a subset of documents indexed from 1000-5000 and have at least 500 characters. The test set is not included in the original corpus, but it has also been downloaded form Milliyet's public website [3].

The data used in this study have been analyzed with the morphological analyser described in Oflazer (1993) and disambiguated with Sak et al. (2007)'s morphological disambiguator. The data have been manually labelled for training and test. The annotated data is made available for pub-

[1] It has only one prefix for intensifying adjectives and adverbs (**sım**sıcak: very hot). It is just a modified version of the first syllable of the original word and also it is not common. There are other prefixes adopted from foreign languages such as **a**normal (abnormal), **anti**sosyal (antisocial) or **na**mert (not brave).

[2] Thanks to Kemal Oflazer for letting us use the corpus

[3] http://www.milliyet.com.tr

Proceedings of the ACL Student Research Workshop, pages 1–8,
Sofia, Bulgaria, August 4-9 2013.

lic use [4]. By making our manually annotated data available, we hope to contribute to future work in this area.

The rest of the paper is organized as follows. Section 2 briefly describes the classification methods used, section 3 explains how these methods are used in implementation and finally the paper is concluded with experimental results.

2 Background

Supervised and unsupervised methods have been used for text classification in different languages (Amasyalı and Diri, 2006; Beil et al., 2002). Among these are Naive Bayes classification (McCallum and Nigam, 1998; Schneider, 2005), decision trees (Johnson et al., 2002) , neural networks (Ng et al., 1997), k-nearest neighbour classifiers (Lim, 2004) and support-vector machines (Shanahan and Roma, 2003).

Bag-of-words model is one of the more intuitive ways to represent text files in text classification. It is simple, it ignores syntax, grammar and the relative positions of the words in the text (Harris, 1970). Each document is represented with an unordered list of words and each of the word frequencies in the collection becomes a feature representing the document. Bag-of-words approach is an intuitive way and popular among document classification tasks (Scott and Matwin, 1998; Joachims, 1997).

Another way of representing documents with term weights is to use term frequency - inverse document frequency (Sparck Jones, 1988). TFIDF is another way of saying that a term is valuable for a document if it occurs frequently in that document but it is not common in the rest of the collection. TFIDF score of a term **t** in a document **d** in a collection **D** is calculated as below:

$$tfidf_{t,d,D} = tf_{t,d} \times idf_{t,D}$$

$tf_{t,d}$ is the number of times **t** occurs in **d** and $idf_{t,D}$ is the number of documents in **D** over the number of document that contain **t**.

The idea behind bag of words and TFIDF is to find a mapping from words to numbers which can also be described as finding a mathematical representation for text files. The output is a matrix representation of the collection. This is also called vector space model representation of the collection in which we can define similarity and distance metrics for documents. One way is to use dot product since each document is represented as a vector (Manning et al., 2008). A number of different dimensions in vector spaces are compared in this study to find the optimal performance.

[4] http://www.ceng.metu.edu.tr/ burakkerim/text_cat

2.1 Morphology

Languages such as Turkish, Czech and Finnish have more complex morphology and cause additional difficulties which requires special handling on linguistic studies compared to languages such as English (Sak et al., 2007). Morphemes may carry semantic or syntactic information, but morphological ambiguity make it hard to pass this information on to other level in a trivial manner especially for languages with productive morphology such as Turkish. An example of possible morphological analyses of a single word in Turkish is presented in Table 1.

alın+Noun+A3sg+Pnon+Nom (forehead)
al+AdjˆDB+Noun+Zero+A3sg+P2sg+Nom (your red)
al+AdjˆDB+Noun+Zero+A3sg+Pnon+Gen (of red)
al+Verb+Pos+Imp+A2pl ((you) take)
al+VerbˆDB+Verb+Pass+Pos+Imp+A2sg ((you) be taken)
alın+Verb+Pos+Imp+A2sg ((you) be offended)

Table 1: Morphological analysis of the word "alın" in Turkish with the corresponding meanings.

We aim to examine the effects of morphological information in a bag-of-words model in the context of text classification. A relevant study explores the prefixing versus morphological analysis/stemming effect on information retrieval in Can et al. (2008). Several stemmers for Turkish are presented for the indexing problem for information retrieval. They use Oflazer's morphological analyzer (Oflazer, 1993), however, they do not use a disambiguator. Instead they choose the most common analysis among the candidates. Their results show that among the fixed length stemmers 5-character prefix is the the best and the lemmatizer based stemmer is slightly better than the fixed length stemmer with five characters. However, they also note that the difference is statistically insignificant. We use Sak et al. (2007)'s disambiguator which is reported with a 96.45% accuracy in their study and with a 87.67% accuracy by Eryiğit (2012)

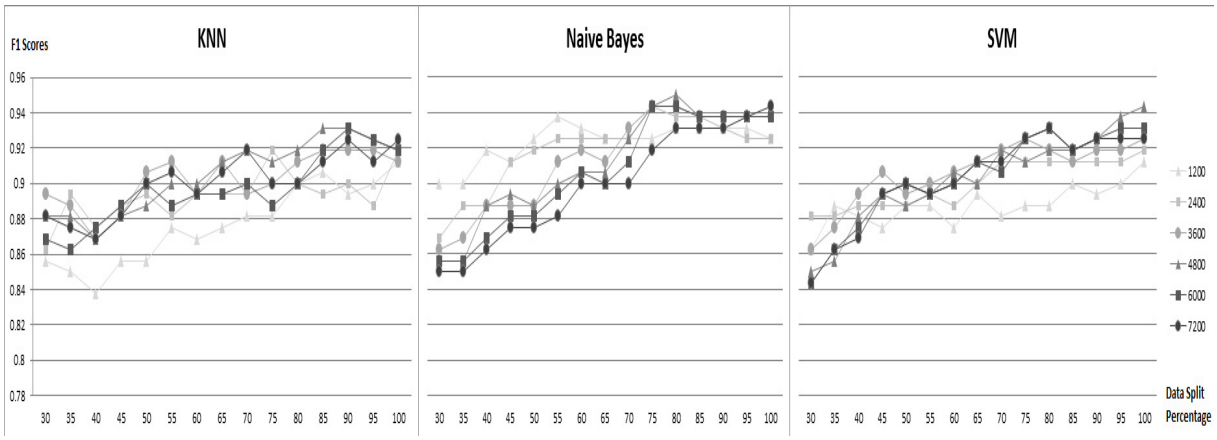


Figure 1: Learning curves with first five characters

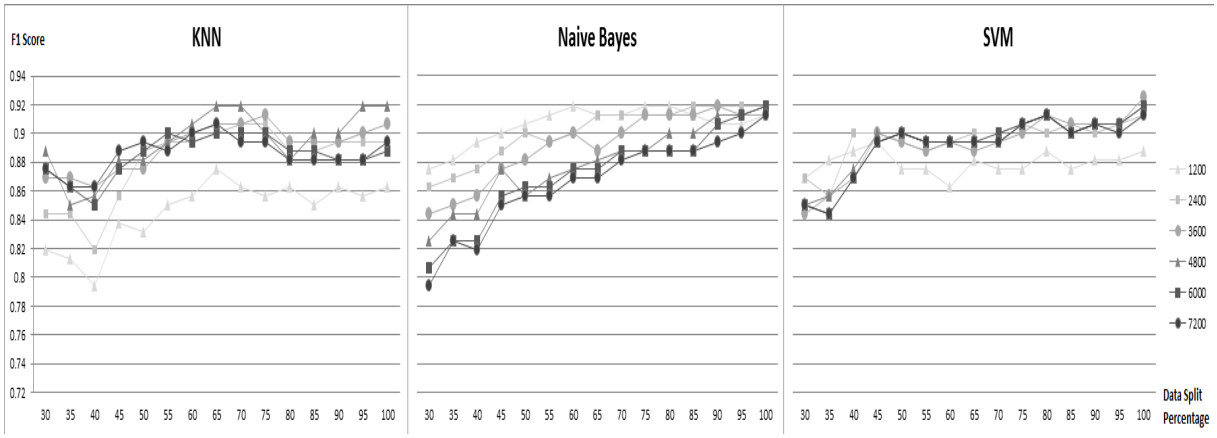


Figure 2: Learning curves with stems

3 Implementation

In the first setting, up to first N characters of each word is extracted as the feature set. A comparison between 4, 5, 6 and 7 characters is performed to choose the best N. In the second setting we use morphological analysis. Each word in documents is analysed morphologically with morphological analyser from Oflazer (1993) and word stems are extracted for each term. Sak's morphological disambiguator for Turkish is used at this step to choose the correct analysis (Sak et al., 2007). Stems are the primary features used for classification. Finally, we add word categories from this analysis as features as POS tags.

We compare these settings in order to see how well morphological analysis with disambiguation performs against a simple baseline of fixed length stemming with a bag-of-words approach. Both stem bags and the first N-character bags are transformed into vector space with TFIDF scoring. Then, different sizes of feature space dimensions are used with ranking by the highest term frequency scores. A range of different dimension sizes from 1200 to 7200 were experimented on to find the optimal dimension size for this study (Table 2). After the collection is mapped into vector space, several learning algorithms are applied for classification. K-Nearest neighbours was implemented with weighted voting of 25 nearest neighbours based on distance and Support Vector Machine is implemented with linear kernel and default parameters. These methods are used with Python, NLTK (Loper and Bird, 2002) and Sci-Kit (Loper and Bird, 2002; Pedregosa et al., 2011).

Training data contains 872 articles labelled and divided into four categories as follows: 235 articles on politics, 258 articles about social news such as culture, education or health, 177 articles on economics and 202 about sports. This data are generated using bootstrapping. Documents are hand annotated with an initial classifier that is trained on a smaller set of hand labelled data. Classifier is used on unknown sam-

ples, then the predictions are manually checked to gather enough data for each class. Test data consists of 160 articles with 40 in each class. These are also manually labelled.

4 Experiments

Experiments begin with searching the optimal prefix length for words with different classifiers. After that, stems are used as features and evaluated with the same classifiers. Section 4.3 contains the comparison of these two features. Finally, morphological information is added to these features and the effects of the extra information is inspected in Section 4.4 .

4.1 Optimal Number of Characters

This experiment aims to find out the optimal prefix length for the first N-character feature to represent text documents in Turkish. We conjecture that we can simulate stemming by taking a fixed length prefix of each word. This experiment was performed with all of the 872 training files and 160 test files. Table 2 shows the results of the experiments where columns represent the number of characters used and rows represent the number of features used for classification.

The best performance is acquired using the first five characters of each word for TFIDF transformation for all classifiers. Can et al. (2008) also reported that the five character prefix in the fixed length stemmer performed the best in their experiments. Learning curves for 5-character prefixes are presented in Figure 1. Although, SVM performs poorer on average compared to Naive Bayes, their best performances show no significant statistical difference according to McNemar's Test. On the other hand, kNN falls behind these two on most of the configurations.

4.2 Stems

Another experiment was conducted with the word stems extracted with a morphological analyser and a disambiguator (Sak et al., 2007). kNN, Naive Bayes and SVM were trained with different feature sizes with increasing training data sizes. The learning curves are presented in Figure 2.

Naive Bayes performs best in this setting even with a small feature set with few training samples. When the corpus size is small, using less features gives better results in SVM and Naive Bayes. As the number of features used in classification increases, the number of samples needed for an adequate classification also increases for Naive Bayes. The performance of SVM also increases with the number of data used in training. More documents leave space for repetitions for stop words and common less informative words an their TFIDF scores decrease and the get less impact on the classification while informative words in each category get relatively higher scores, therefore an increase in data size also increases performance. As the training size increases feature space dimension becomes irrelevant and the results converge to a similar point for Naive Bayes. On the other hand, 1200 features are not enough for kNN and SVM. With larger feature sets kNN and SVM also give similar results to Naive Bayes although kNN is left behind especially with less number of features since it directly relies on the similarity based on these features in vector space and most of them are same in each document since we choose them with term frequency.

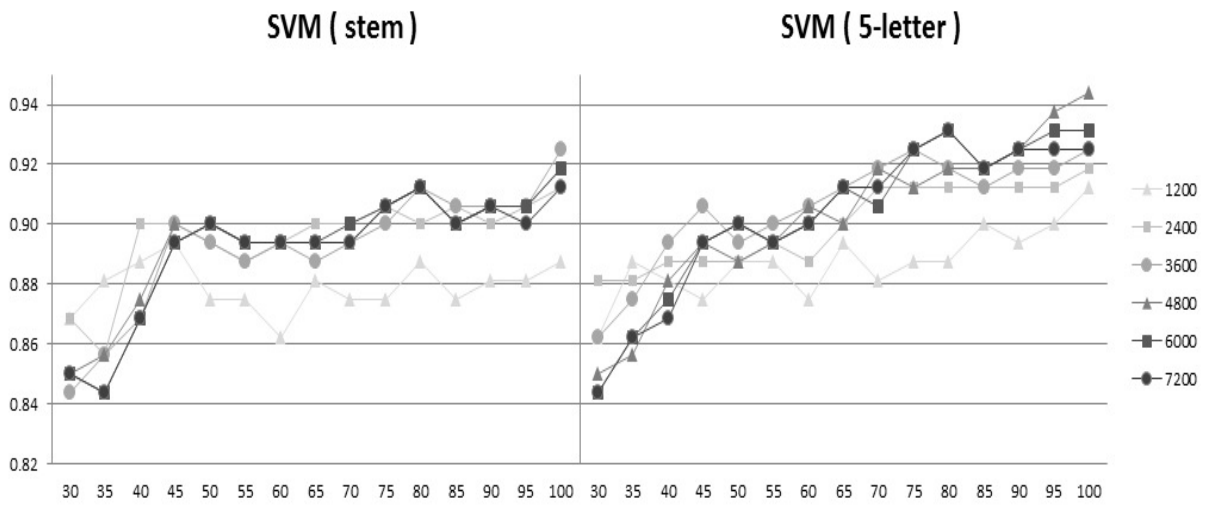
4.3 5-Character Prefixes vs Stems

This section provides a comparison between two main features used in this study with three different classifiers. F1 scores for the best and worst configurations with each of the three classifiers are presented in Table 3. Using five character prefixes gives better results than using stems. Naive Bayes with stems and five character prefixes disagree only on six instances out of 160 test instances with F1 scores of 0.92 and 0.94 respectively in the best configurations. There is no statistically significant difference.

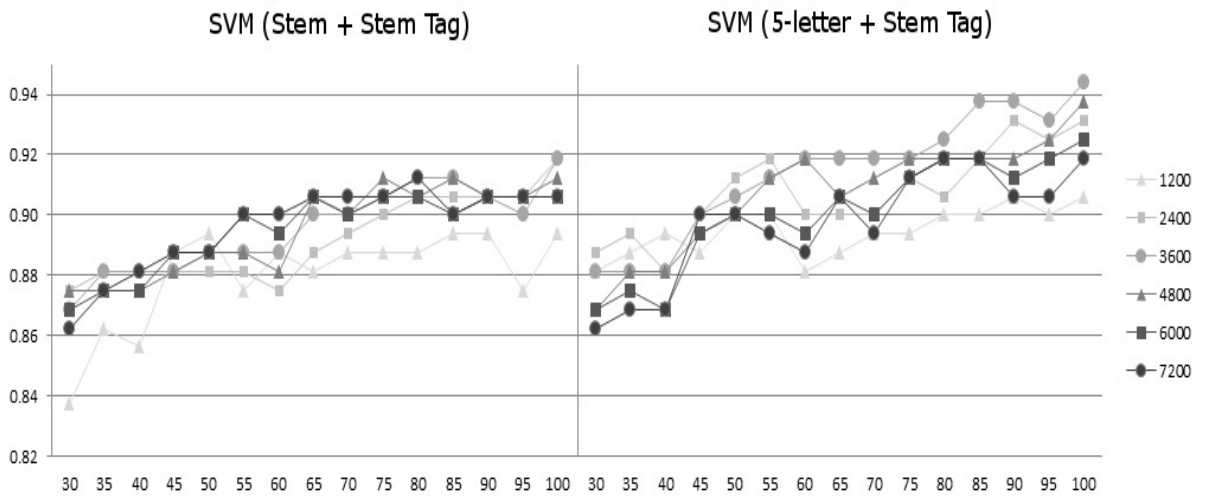
Similarly, results for SVM with stems for the best and the worst configurations is considered to be not statistically significant. McNemar's Test (McNemar, 1947) is shown to have low error in detecting a significant difference when there is none (Dietterich, 1998).

	Worst		**Best**	
	First 5	**Stems**	**First 5**	**Stems**
KNN	91.250	86.875	92.500	91.875
NB	92.500	91.250	94.375	91.875
SVM	91.250	88.750	93.175	92.500

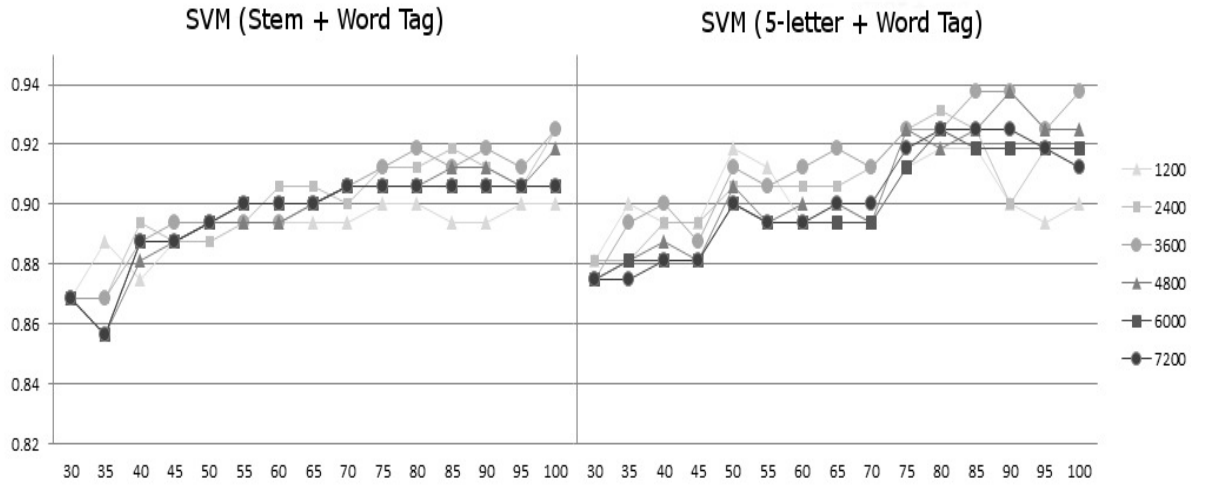
Table 3: Comparison of F1-scores for best and worst results in each classifier with each feature.



(a) Learning curves without tags



(b) Learning curves with stem tags



(c) Learning curves with word tags

Figure 3: Learning curves for SVM

	KNN				NB				SVM			
	4	**5**	**6**	**7**	**4**	**5**	**6**	**7**	**4**	**5**	**6**	**7**
1200	90.00	91.25	86.87	84.37	93.12	92.50	93.12	90.00	89.37	91.250	90.62	88.75
2400	89.37	91.25	87.50	86.62	89.37	91.25	87.50	86.62	90.62	91.87	90.00	88.12
3600	86.87	91.25	90.00	88.17	93.75	93.75	92.50	91.87	90.62	91.87	90.00	88.12
4800	90.00	91.87	91.25	88.17	93.12	93.75	91.87	91.25	90.62	91.87	90.00	88.12
6000	88.75	91.87	91.87	90.62	92.50	93.75	92.50	90.62	90.62	**93.12**	**93.12**	90.00
7200	89.37	**92.50**	91.25	89.37	90.62	**94.37**	91.87	91.25	90.62	92.50	91.25	90.62

Table 2: F1-scores with different prefix lengths and dimensions.

4.4 SVM with POS Tags

The final experiment examines the effects of POS tags that are extracted via morphological analysis. Two different features are extracted and compared with the base lines of classifiers with stems and first five characters without tags. Stem tag is the first tag of the first derivation and the word tag is the tag of the last derivation and example features are given in Table 4. Since derivational morphemes are also present in the morphological analyses word tags may differ from stem tags. In addition, words that are spelled in the same way may belong to different categories or have different meanings that can be expressed with POS tags. Al+Verb (take) and Al+Adj (red) are different even though their surface forms are the same.

Analysis	al+Adjˆ DB+Noun+Zero+ A3sg+Pnon+Gen (of red)
First 5 characters.	alın (of red, forehead, (you) be taken, (you) be offended ...)
Stem	al (red, take)
Stem + Stem Tag	al+Adj (red)
Stem + Word Tag	al+Noun (red)

Table 4: Example features for word "alın".

Using POS tags with stems increases the success rate especially when the number of features is low. However, using tags of the stems does not make significant changes on average. The best and the worst results differ with baseline with less than 0.01 points in F1 scores as seen in Figure 3. This may be due to the fact that the same stem has a higher chance of being in the same category even though the derived final form is different. Even though, this may add extra information to the stems, results show no significant difference. Adding stem or word tags to the first five characters increases the success when the number of training instances are low, however, it has no significant effect on the highest score. Using tags with five characters has positive effects when the number of features are low and negative effects when the number of features are high.

5 Conclusion

In this study, we use K-Nearest Neighbours, Naive Bayes and Support Vector Machine classifiers for examining the effects of morphological information on the task of classifying Turkish news articles. We have compared their performances on different sizes of training data, different number of features and different feature sets. Results suggest that the first five characters of each word can be used for TFIDF transformation to represent text documents in classification tasks. Another feature used in the study is word stems. Stems are extracted with a morphological analyser which is computationally expensive and takes a lot of time compared to extracting first characters of a word. Although different test sets and training data may change the final results, using a simple approximation with first five characters to represent documents instead of results of an expensive morphological analysis process gives similar or better results with much less cost. Experiments also indicate that there is more place for growth if more training data is available as most of the learning curves presented in the experiments point. We particularly expect better results with POS tag experiments with more data. Actual word categories and meanings may differ and using POS tags may solve this problem but sparsity of the data is more prominent at the moment. The future work includes repeating these experiments with larger data sets to explore the effects of the data size.

References

Charu C. Aggarwal and Philip S. Yu. 2000. Finding generalized projected clusters in high dimensional spaces. *SIGMOD Rec.*, 29(2):70–81.

M. Fatih Amasyalı and Banu Diri. 2006. Automatic Turkish text categorization in terms of author, genre and gender. In *Proceedings of the 11th international conference on Applications of Natural Language to Information Systems*, NLDB'06, pages 221–226, Berlin, Heidelberg. Springer-Verlag.

Florian Beil, Martin Ester, and Xiaowei Xu. 2002. Frequent term-based text clustering. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 436–442, New York, NY, USA. ACM.

Fazlı Can, Seyit Koçberber, Erman Balçık, Cihan Kaynak, H. Çağdaş Öcalan, and Onur M. Vursavaş. 2008. Information retrieval on turkish texts. *JASIST*, 59(3):407–421.

Ruket Çakıcı and Jason Baldridge. 2006. Projective and non-projective Turkish parsing. In *Proceedings of the 5th International Treebanks and Linguistic Theories Conference*, pages 43–54.

Özlem Çetinoğlu and Kemal Oflazer. 2006. Morphology-syntax interface for Turkish LFG. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 153–160, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923.

Gülşen Eryiğit. 2012. The impact of automatic morphological analysis & disambiguation on dependency parsing of turkish. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey, 23-25 May.

Gülşen Eryiğit, Joakim Nivre, and Kemal Oflazer. 2008. Dependency parsing of Turkish. *Comput. Linguist.*, 34(3):357–389, September.

George Forman. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, March.

Dilek Z. Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. 2000. Statistical morphological disambiguation for agglutinative languages. In *Proceedings of the 18th conference on Computational linguistics - Volume 1*, COLING '00, pages 285–291, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zelig Harris. 1970. Distributional structure. In *Papers in Structural and Transformational Linguistics*, pages 775–794. D. Reidel Publishing Company, Dordrecht, Holland.

M. Ikonomakis, S. Kotsiantis, and V. Tampakas. 2005. Text classification: a recent overview. In *Proceedings of the 9th WSEAS International Conference on Computers*, ICCOMP'05, pages 1–6, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS).

Thorsten Joachims. 1997. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 143–151, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

D. E. Johnson, F. J. Oles, T. Zhang, and T. Goetz. 2002. A decision-tree-based symbolic rule induction system for text categorization. *IBM Syst. J.*, 41(3):428–437, July.

Heui-Seok Lim. 2004. Improving kNN based text classification with well estimated parameters. In Nikhil R. Pal, Nikola Kasabov, Rajani K. Mudi, Srimanta Pal, and Swapan K. Parui, editors, *Neural Information Processing, 11th International Conference, ICONIP 2004, Calcutta, India, November 22-25, 2004, Proceedings*, volume 3316 of *Lecture Notes in Computer Science*, pages 516–523. Springer.

Tao Liu, Shengping Liu, and Zheng Chen. 2003. An evaluation on feature selection for text clustering. In *In ICML*, pages 488–495.

Edward Loper and Steven Bird. 2002. Nltk: the natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *Proceesings of the Workshop on learning for text categorization, AAAI'98*, pages 41–48.

Quinn McNemar. 1947. Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. *Psychometrika*, 12(2):153–157.

Hwee Tou Ng, Wei Boon Goh, and Kok Leong Low. 1997. Feature selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '97, pages 67–73, New York, NY, USA. ACM.

Kemal Oflazer. 1993. Two-level description of Turkish morphology. In *Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics*, EACL '93, pages 472–472, Stroudsburg, PA, USA. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2007. Morphological disambiguation of Turkish text with perceptron algorithm. In *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '07, pages 107–118, Berlin, Heidelberg. Springer-Verlag.

Karl-Michael Schneider. 2005. Techniques for improving the performance of naive bayes for text classification. In *In Proceedings of CICLing 2005*, pages 682–693.

Sam Scott and Stan Matwin. 1998. Text classification using WordNet hypernyms. In *Workshop: Usage of WordNet in Natural Language Processing Systems, ACL'98*, pages 45–52.

James G. Shanahan and Norbert Roma. 2003. Boosting support vector machines for text classification through parameter-free threshold relaxation. In *Proceedings of the twelfth international conference on Information and knowledge management*, CIKM '03, pages 247–254, New York, NY, USA. ACM.

Karen Sparck Jones. 1988. A statistical interpretation of term specificity and its application in retrieval. In Peter Willett, editor, *Document retrieval systems*, pages 132–142. Taylor Graham Publishing, London, UK, UK.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 42–49, New York, NY, USA. ACM.

Yiming Yang and Jan O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.