Classification of Jewish Law Articles According to the Ethnic Group of their Writers Using Stems Only

Research paper 168

Abstract. In this paper, we investigate automatic classification of one thousand Jewish Law articles written in Hebrew-Aramaic according to the ethnic group of their authors. After extracting the stems of the words in each article, the most frequent (>95%) and the least frequent (<5%) stems were removed. Using 480 stems as inputs to an artificial neural network model, the classification result, 85% of the validation examples, is reasonable, considering that the stemming software accuracy is not perfect. Discarding 340 less relevant stems, and retraining with 140 stems gave the same error rate. It seems that classification based on stems only may suitable for such a classification of other texts. It will be interesting to check whether stylistic classification can be also used for other tasks of ethnic classification, e.g.: Sunni Muslims and Shia Muslims; Protestant Christians and Catholic Christians.

1 Introduction

Text classification (TC) is the supervised learning task of assigning natural language text documents to one or more predefined classes (also called categories or topics) according to their content. The meaning of supervised in this definition is that all the documents in a training set are pre-assigned a class before the training process starts.

The beginning of research in TC can be identified with Maron's work on probabilistic text classification [26]. Nowadays, TC is applied in many domains, such as: document indexing, document filtering, and word sense disambiguation. Moreover, current-day TC presents challenges due to the large number of features present in the text set, their dependencies and the large number of training documents. Research in this domain can contribute to research in other related domains, e.g.: clustering, information retrieval (IR), information extraction (IE), machine learning (ML), natural language processing (NLP), Word sense disambiguation (WSD), text filtering, and text mining [20], [31].

One of the machine learning methods employed for text classification is the artificial neural networks (ANN) technique [29], which was found superior to some other ML techniques [13]. ANN modeling was recently employed for predicting the importance of a literature abstract to researchers downloading references, using the stemmed English words in the abstract as inputs to the ANN [23]. Thus it was interesting to see if this method can be also applied to Hebrew-Aramaic texts.

In our research, we plan and apply a model that classifies Responsa (letters written in response to legal questions) according to the ethnic group of their writers. Our corpus is a

collection of Responsa written in Hebrew-Aramaic by a number of rabbinic scholars, which are authorities in Jewish law.

Our plan is to check whether we can succeed in such a task using only stems of words, excluding very frequent stems and very rare stems. We have built an artificial neural network (ANN) for implementing this task.

The structure of this paper is as follows. First we describe the basics of the Hebrew-Aramaic languages word structure relevant to the task of TC; then a brief introduction to the ANN modeling is presented, with a more detailed view of the particulate large-scale ANN algorithms that we used for the TC task; the data set we used will be presented, with the pre-processing technique we have employed. Results of the ANN classification will be shown, and future avenues of research will conclude the paper.

2 The Hebrew and the Aramaic Languages

2.1 The Hebrew Language

Hebrew is a Semitic language. It is written from right to left. Hebrew texts present special problems: (1) function words tend to be conflated into word affixes in Hebrew, thus decreasing the number of function words but increasing the amount of morphological features that can be exploited and (2) the richness of Hebrew morphology (more details are given below).

Hebrew words in general and Hebrew verbs in particular are based on three (sometimes four) basic letters that create the word's stem.

Except for the word's stem, there are other several components which create the word's declensions, e.g.: conjugations, verb types, subject, prepositions, belonging, object and terminal letters. In Hebrew, it is impossible to find the declensions of a certain stem without an exact morphological analysis based on the features mentioned above.

The English language is richer in its vocabulary than Hebrew. The English language has about 40,000 stems while Hebrew has only about 3,500 and the number of lexical entries in the English dictionary is 150,000 compared with only 35,000 in the Hebrew dictionary [1].

However, the Hebrew language is richer in its morphology forms. The Hebrew language has 70,000,000 valid (inflected) forms while English has only 1,000,000. For example, the single Hebrew word *vkhsykhvhv* (וכשיכוהו) is translated into the following sequence of six English words: "and when they will hit him". In comparison to the Hebrew verb which undergoes a few changes the English verb stays the same.

In Hebrew, there are up to seven thousand declensions for only one stem, while in English there is only a few declensions. For example, the English word eat has only four declensions (eats, eating, eaten and ate). The relevant Hebrew stem *'khl* (אכל,"eat") has thousands of declensions. Ten of them are presented below: (1) *'khlty* (אכלתי, "I ate"), (2) 'khlt (אכלת, "you ate"), (3) *'khlnv* (אכלנו, "we ate"), (4) *'khvl* (אוכל, "he eats"), (5) *'khvlym*

(אוכלים, “they eat”), (6) *‘tkhl* (תאכל, “ she will eat”), (7) *l‘khvl* (לאכל, “to eat”), (8) *‘khltyv* (אכלתיו, “I ate it”), (9) *v‘khlty* (ואכלתי, “and I ate”) and (10) *ks‘khlt* (כשאכלת, “when you ate”).

2.2 The Aramaic Language

Aramaic is another Semitic language. The term Aramaic is derived from Aram, the fifth son of Shem, the firstborn of Noah. [Gen. 10:22]. It is particularly closely related to Hebrew, and was written in a variety of alphabetic scripts. (What is usually called "Hebrew" script is actually an Aramaic script). Aramaic was the language of Semitic peoples throughout the ancient Near East. It is spoken for at least three thousand years. Aramaic is still spoken today in its many dialects, especially among the Chaldeans and Assyrians. [35]. In the Bible, there are large sections of Aramaic texts in the books of Daniel and Ezra and odd words in other books. Aramaic has influenced Hebrew (as French has influenced English) in words, phrases and grammar.

Although Aramaic and Hebrew have much in common, there are several major differences between them. The main difference in grammar is that while Hebrew uses aspects and word order to create tenses, Aramaic uses tense forms. Another important difference is that there are several types of changes in one particular letter in many words. For instance: (1) in some cases an Hebrew prefix is replaced in Aramaic by a suffix (e.g. the Hebrew prefix ה is changed into the Aramaic suffix א) (2) the Hebrew plural noun suffixes ות and ים, are changed into ין and נא in Aramaic and (3) the word *which* that is integrated as the prefix ש in Hebrew is changed into ד in Aramaic.

3 Previous Stylistic Classification of Hebrew-Aramaic Texts

CHAT, a system for stylistic classification of Hebrew-Aramaic texts is presented in [29], [24], [25]. CHAT present applications of several TC tasks to Hebrew-Aramaic texts:

1. Which of a set of known authors is the most likely author of a given document of unknown provenance?
2. Were two given corpora written/edited by the same author or not?
3. Which of a set of documents preceded which and did some influence others?
4. From which version (manuscript) of a document is a given fragment taken?

CHAT uses as features only single words, prefixes and suffixes. This system uses simple ML methods such as Winnow and Perceptron. Its datasets contain a few hundreds of documents. CHAT does not investigate the classification of responsa according to the ethnic group of their authors.

Classification of Biblical documents has been done by Radai [32], [34], [35]. However, he did not implement any ML method.

4 A Brief Introduction on Artificial Neural Networks Modeling

ANN modeling is done by learning from examples. ANN is a network of simple (sigmoid, for example) mathematical "neurons" connected by adjustable weighted links. The most used ANN architecture is feed-forward two-layer ANN, in which neurons are placed in one hidden layer between the data inputs and the neurons of the output layer, and the information flows only from the inputs to the hidden neurons and from them to the output neurons. Training examples are presented as inputs to the ANN, which uses a "teacher" to train the model. An error is defined as the difference between the model outputs and the known "teacher" outputs. Error back-propagation algorithms adjust the initial random-valued model connection weights to decrease the error, by repeated presentations of input vectors [40], [38], [41], [3]. Once the ANN is trained and verified by presenting inputs not used in the training, the ANN is used to predict outputs of new inputs presented to it.

There are several obstacles in applying an ANN to systems containing a large number of inputs and outputs. Most ANN training algorithms need thousands of repeated presentations ("epochs") of the inputs to finally achieve small modeling errors. Large ANN tends to get stuck in local minima during the training.

An efficient training algorithm set, developed by Guterman and Boger [19], [5], can easily train large scale ANN models, as it pre-computes non-random initial connection weights from the manipulation of training data sets, avoiding or escaping local minima during the training. The ANN architecture used by the Guterman-Boger algorithm is the most common one - fully connected forward only, one hidden layer, and sigmoid activation function. The Guterman-Boger algorithm was successfully used to train ANN models with hundreds to thousands of inputs and outputs [4], [18], [7], [8].

In real-life models, not all inputs are influencing the model outputs in the same degree. A knowledge extraction technique is the ranking of the inputs according to their relevance to the ANN prediction accuracy. Calculating the relative contribution of each input to the variance in the hidden neurons inputs when the training set is presented to the trained ANN model does this. A low relative contribution means that either the variance of the input is small, or that the ANN training has assigned low connection weights from that input to all hidden neurons [5]. The detailed derivation of the input relevance calculation is given in [10]. The least relevant inputs may be discarded and the ANN can be re-trained with the reduced input set that usually gives better prediction accuracy. The explanations for this possible improvement are: a) Elimination of noise or conflicting data in the non-relevant inputs; b) Reduction of the number of connection weights in the ANN that improves the ratio of the number of examples to the number of connection weights, thus reducing the chance of over-fitting small number of examples to a model with many parameters ("over-training").

5 The Application of ANN to Text Classification

The idea to match the capabilities of ANN modeling to information retrieval is not new, and many papers are dealing with it. Most of the papers use the unsupervised self-organized maps (SOM) technique for grouping similar examples into clusters [21]. Thus text clusters are formed based on the similarity of keywords in the texts. Once trained, the ANN will classify new documents as belonging to one of these clusters. [36], [37], [42], [28]. Recent reviews discuss ANN along with other "soft" tools for Web mining application [30] and text classification [39]. Several text classification algorithms were compared, and ANN modeling was found to be superior [13]. One of us has used ANN to predict the importance of an e-mail message, or the relevance of a downloaded paper abstract to a researcher [9], [23].

The ability of the ANN to model non-linear, non-obvious relationships can be applied to the matching of the textual features (inputs to the ANN) to the user relevance rating (ANN outputs). When applying statistical methods for the required modeling, subjective selections of the number of terms and the form of the model equations are made. No such assumptions are needed in ANN modeling.

In order to use a classification mechanism such as an ANN for document filtering, an appropriate document representation is required. In our case we used a binary vector representation of terms to represent the documents.

6 Our Model

Our model, in general, is composed of the six following steps (several steps will be explained below):

(1) Building a data set composed of various Jewish Law articles.
(2) For each article transform each word (excluding stop-list words) into its estimated stem using a stem learning program.
(3) Represent each document as vector of its stems
(4) Stems in the bottom 5% and top 95% count were discarded
(5) Apply the ANN on these stems
(6) Analyze the trained ANN model to identify the more relevant stems
(7) Reduce the stem set
(8) Re-apply the ANN on the reduced set of stems

At step (2) we applied a program that proposes an estimated stem for any given word (without its context) written either in Hebrew or in Aramaic [Daya et al, 2004; Daya 2005]. This program is based on WINNOW (a simple ML method), identifies the correct stem in about 80% chance of success. It produces only stems made up of three letters. That is, it doesn't find the correct stems for words that their stems contain more than three letters.

7 Experiments

The dataset employed contained 1000 reponsa collected from 20 different rabbinic books, 500 written by Sepharadic Jews and 500 written by Askenazic Jews. Although this data set is relatively small, it is important to point out that these responsa are hard to obtain, because usually they are not available online. These responsa were downloaded from The Global Jewish Database (The Responsa Project[1]) at Bar-Ilan University.
The total number of words in all the files was 2,278,683. After reducing stop-list words, abbreviations and words that contain only one letter, the total number of words in all the files was 1,043,550.These words were transformed to 887 different 3-letter stems using the stem-program mentioned above.

For the ANN modeling, stems in the bottom 5% and top 95% count were discarded, and the rest were used to form a binary vector, where 1 signifies the presence of a stem in the text. The number of different legal stems with frequency in files between 5% <--> 95% was 480, the number of stems that were removed, was 407. Thus, an ANN model was trained with the term presence vector as input, and with five hidden neurons and two binary outputs. The ANN target for a document is a 2-bit binary vector with 1 at the Sepharadic position or 1 at the Ashkenazic position.

The data was partitioned by a random selection into 701 training set and 299 validation set, not used in the training. The ANN was trained with the Guterman-Boger set of algorithms described in the earlier sections. The trained ANN model was analyzed for identifying the more relevant inputs that were used to train another, smaller, ANN.

The ANN modeling, using 480 inputs, 5 hidden neurons and 2 outputs architecture, gave zero errors on the training set, 15.4% errors on the validation set. Analysis of the trained ANN model identified 140 stems as the more relevant ones. Retraining an ANN with these inputs, gave a slightly better error rate, 15.0%.

These 140 stems appear to be the most significant for classifying Jewish Law articles according to the Ethnic group of their writers since they have different distribution between the two Ethnic groups. In contrast, the 340 removed stems have the same distribution between the two Ethnic groups.

Among the 140 stems that found to support the classification task we find a few Aramaic stems that are more common in use of Ashkenazic Jews, e.g.: (1) **כוי** that stands as a stem for the word **כוותיהו** and (2) **פקע** that stands as a stem for the word **אפקעינן.** Examples for stems that are more common in use of Sephardic Jews **are:** (1) **מור** that stands as a stem for the word **מרן** (which is a pen name for one of the most important Sephardic Rabies) and (2) **צדק** that stands as a stem for the word **צדיק.**

[1] http://www.biu.ac.il/ICJI/Responsa/index.html

Among the 340 removed stems on the one side we can find rather frequent stems such as: (1) **למד** that stands as a stem for the family of words related to the Hebrew word **למד** (learn) and (2) דבר that stands as a stem for the family of words related to the Hebrew word **דבר** (talk). On the other side we can find non-frequent stems such as: (1) נגח that stands as a stem for the family of words related to the Hebrew word נגח (gore) and (2) נגנ that stands as a stem for the family of words related to the Hebrew word **נגנ** (to play music).

The 85% correct classification result is reasonable. A possible explanation to this finding might be that classification based on stems depends on the efficiency of the stemming program to correctly represent the words.

8 Conclusions and Future Work

Stem-based classification although its simplicity has been found as rather successful for ethnic classification of responsa written in Hebrew-Aramaic.

Future directions for research are: (1) Conducting more experiments using additional Hebrew-Aramaic documents from additional domains, (2) Checking whether stem-based classification can be also used for other tasks of ethnic classification, e.g.: Sunni Muslims and Shia Muslims; Protestant Christians and Catholic Christians and (3) It will be interesting to compare our research to the same classification task based on more complex feature such as words and/or linguistic features.

Concerning research on additional ethnic groups, there are many additional potential directions. For example: (1) Which baseline methods are good for which classification tasks? (2) What are the specific reasons for methods to perform better or worse on different classification tasks? (3) What are the guidelines to choose the correct methods for a certain classification task?

Acknowledgements. The authors would like to thank Ezra Daya, Dan Roth and Shuly Wintner for letting us to apply their stem program.

References

1. Argamon-Engelson, S., M. Koppel, G. Avneri (1998). Style-based text categorization: What newspaper am I reading?, in Proc. of AAAI Workshop on Learning for Text Categorization, 1998, pp. 1-4.
2. Baayen, H., H. van Halteren, F. Tweedie (1996). Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution, Literary and Linguistic Computing, 11, 1996.

3. Bishop, C.M. (1995). *Neural Networks for Pattern Recognition.* Clarendon Press.
4. Boger Z. (1992). Application of neural networks to water and wastewater treatment plant operation. *Transactions of the Instrument Society of America*, 31 (1), 25-33.
5. Boger, Z. & Guterman, H. (1997). Knowledge extraction from artificial neural networks models. *Proc. of the IEEE Intl. Conference on Systems Man and Cybernetics, SMC'97*, Orlando, Florida, 3030-3035.
6. Boger, Z., Kuflik, T., Shoval P. & Shapira, B. (2001). Automatic keyword identification by artificial neural networks compared to manual identification by users of filtering systems. *Information Processing & Management*, 37 (2) 187-198.
7. Boger, Z. (2002). Who is afraid of the Big Bad ANN? *Proc. of the International Joint Conference on Neural Networks*, IJCNN'02, Hawaii, 2000-2005.
8. Boger, Z. (2003). Selection of quasi-optimal inputs in chemometrics modeling by artificial neural network analysis. *Analytica Chimica Acta*, 490, (1-2), 31-40.
9. Boger, Z., Kuflik, T., Shoval P. & Shapira, B. (2001). Automatic keyword identification by artificial neural networks compared to manual identification by users of filtering systems. *Information Processing & Management*, 37 (2) 187-198.
10. Boger, Z. (2003). Selection of quasi-optimal inputs in chemometrics modeling by artificial neural network analysis. *Analytica Chimica Acta*, 490, (1-2), 31-40.
11. Choueka, Y., Conley E. S. and Dagan I., A comprehensive bilingual word alignment system: application to disparate languages - Hebrew and English, in J. Veronis (Ed.), Parallel Text Processing, Kluwer Academic Publishers, 2000, pp. 69–96.
12. Clack, C., Farringdon, J., Lidwell, P., And Yu, T. 1997. Autonomous document classification for business. In Proceedings of the 1st International Conference on Autonomous Agents (Marina del Rey, CA, 1997), 201–208.
13. Corrêa, R.F. and Ludermir, T.B., Automatic Text Categorization: Case Study, Proceedings of the VII Brazilian Symposium on Neural Networks (SBRN'02), 2002.
14. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning, 20 (1995) 273-297
15. de Vel, O., A. Anderson, M. Corney and George M. Mohay (2001). Mining e-mail content for author identification forensics. SIGMOD Record 30(4), pp. 55-64
16. Daya, E., Roth D., and Wintner, S. Learning Hebrew Roots: Machine Learning with Linguistic Constraints. *Proceedings of EMNLP'04*, Barcelona, July 2004.
17. Daya, E. Learning to Identify Semitic Roots, Master Thesis, 2005. University of Haifa, Israel.
18. Greenberg, S. & Guterman, H. (1996). Neural networks classifiers for automatic real-world image recognition. *Applied Optics*, 35, 4598-4609.
19. Guterman, H. (1994). Application of principal component analysis to the design of neural networks. *Neural, Parallel and Scientific Computing*, 2, 43-54.
20. Knight, K. 1999. Mining online text. Commun.ACM 42, 11, 58–61.
21. Kohonen, T. (1997). Exploration of very large databases by self-organizing maps. *Proc. of the IEEE International Conference on Neural Networks*, 1, PL1-6.
22. Kuflik, T. (2003). Methods for Definition of Content-Based and Rule-Based User Profiles in Information Filtering Systems, *PhD. Dissertation*. Ben-Gurion University of the Negev.

23. Kuflik, T., Boger, Z., Shoval P. (2006), Filtering search results using an optimal set of terms identified by an artificial neural network, *Information Processing & Management*, (in Press)
24. Koppel, M., Mughaz D. and Schler J. (2004). Text categorization for authorship verification in Proc. 8th Symposium on Artificial Intelligence and Mathematics, Fort Lauderdale, FL, 2004.
25. Koppel, M., Mughaz D. and Akiva N. (in press), New Methods for Attribution of Rabbinic Literature , Hebrew Linguistics: A Journal for Hebrew Descriptive, Computational and Applied Linguistics, to appear.
26. Maron, M. 1961. Automatic indexing: an experimental inquiry. J. Assoc. Comput. Mach. 8, 3, 404–417.
27. Melamed, Rabbi Ezra Zion, Aramaic-Hebrew-English Dictionary, Feldheim, ISBN: 1-58330-776-1, 2005.
28. Merkl, D. and Rauber, A. Document classification with unsupervised artificial neural networks, in Soft Computing in Information Retrieval: Techniques and Applications (F. Crestani and G. Pasi, eds.), vol. 50, pp. 102-121, Heidelberg: Physica Verlag, 2000.
29. Mughaz, D. *Classification Of Hebrew Texts according to Style*, M.Sc. Thesis (in Hebrew), Bar-Ilan University, Ramat-Gan, Israel (2003).
30. Pal, S.K, Talwar, V. & Mitra, P. (2002). Web mining in soft computing framework: relevance, state of the art and future directions. *IEEE Transactions on Neural Networks*, 13 (5) 1163-1177.
31. Pazienza, M. T., ed. 1997. Information Extraction. Lecture Notes in Computer Science, Vol. 1299. Springer, Heidelberg, Germany.
32. Radai, Y. (1978). Hamikra haMemuchshav: Hesegim Bikoret uMishalot (in Hebrew), Balshanut Ivrit 13: 92-99
33. Radai, Y. (1979). Od al Hamikra haMemuchshav (in Hebrew), Balshanut Ivrit 15: 58-59
34. Radai, Y. (1982). Mikra uMachshev: Divrei Idkun (in Hebrew), Balshanut Ivrit 19: 47-52
35. Rosenthal F., Aramaic Studies During the Past Thirty Years, THE JOURNAL OF NEAR EASTERN STUDIES, pp 81-82, Chicago: 1978.
36. Ruiz, M.E. & Srinivasan, P. (1999). Hierarchical neural networks for text categorization. *Proc. of the 22nd Intl. Conference on Research and Development in Information Retrieval*, 281-282.
37. Ruiz, M.E. & Srinivasan, P. (2002). Hierarchical text categorization using neural networks. *Information Retrieval*, 5 (1) 87-118.
38. Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature*, 323, 533-536.
39. Sebastiani, F. (2002). Machine learning in automated text categorization, ACM Computing Surveys 34 (1), pp. 1-47.
40. Werbos, P. (1974). Beyond Regression: New tools for prediction and analysis in the behavioral sciences. *Ph.D. Dissertation*, Committee on Appl. Math., Harvard Univ.
41. Werbos, P. (1993). *Roots of Back-Propagation: From Ordered Derivatives to Neural Networks to Political Forecasting*. John Wiley and Sons, Inc.
42. Wermter, S. (2000). Neural network agents for learning semantic text classification. *Information Retrieval*, 3, 87-103.