K-means Clustering for POS Tagger Improvement

Gabi Rolih

Department of Linguistics and Philology, Uppsala University
Engelska parken, Thunbergsv. 3H, 751 26 Uppsala
gabirolih@gmail.com

1. Introduction

With the emergence of social media, the internet is turning into an enormous openly accessible corpus with diverse language and domain selection that keeps growing. Using this data would bring many benefits to natural language processing tasks. However, the language use on the internet differs greatly from the standard language due to inconsistent capitalization, omission of diacritics, non-standard spelling and colloquial expressions. The currently available tools in natural language processing (NLP) are not created for dealing with such cases and are therefore not adequate for such texts. Much of the latest research was focusing on creating specialized tools that would be able to deal with non-standard language.

2. Goal of the paper

In this article, we present an implementation of K-means clustering on CMC (computer-mediated communication) data in order to further improve a state-of-the-art tagger for Slovenian. We evaluate the tagger against a human annotated dataset and compare the results with previous work.

In section 2, previous work on this problem is presented. In sections 3, 4 and 5, our datasets, tools and implementation are outlined. In section 6 the performance is evaluated and discussed. In section 7 the article is concluded with future work suggestions.

3. Previous work

Researching CMC data has been in focus in the more recent years.

Eisenstein (2013) analyzes how language is used on the internet and how the NLP community typically deals with these problems. He concludes that there are two standard approaches to NLP tasks for CMC data: normalization and domain adaptation. Normalization is converting the non-standard language into a standard language. This includes changing capitalization, punctuation, spelling and in some cases even changing words into their more formal counterparts. Domain adaptation means adapting already existing tools to a new domain, which in this case is the non-standard language.

Ljubešić et al. (2017) run various experiments to adapt the ReLDI tagger, a state-of-the-art tagger for Slovene, to CMC data. They do this by retraining the tagger on CMC data, using an inflectional lexicon, adding normalization data and using clustering information. One approach that yields significant results is creating Brown clusters (Brown et al., 1992) on CMC data and using that to retrain the tagger. For our research we use the same tools and approaches, except that we use K-means clustering technique instead of Brown clusters.

Brown clusters are a popular choice for word clustering because they are efficient and scale well to large datasets. Turian et al. (2010) evaluate Brown clusters, Collobert and Weston embeddings and hierarchical log-linear (HLBL) embeddings for named-entity recognition (NER) and chunking tasks and confirm that Brown clusters perform best.

Owoputi et al. (2013) successfully use Brown clusters to improve PoS tagging in online conversational texts. They construct a state-of-the-art tagger for Twitter and IRC texts with accurracy above 90.

K-means clustering is a simple and very popular clustering algorithm, but it is not very common in NLP tasks. Lin and Wu (2009) attempt to use K-means on word phrases and use them for NER tasks. Their system achieves the best result for NER systems at the time. They argue that using the cluster on phrases rather than words brings better results.

4. K-means clustering

K-means clustering is a technique proposed by MacQueen et al. (1967). It splits the data into a number of clusters based on the proximity of points to the cluster center. The number of clusters K must be provided as input.

K-means is an iterative algorithm, where each iteration consists of two steps. Step one is assigning clusters to points. For each point, K-means calculates the distances from the point to the cluster centroids. The point is then assigned to the closest cluster centroid. When all points are assigned a cluster, K-means proceeds with step 2. In this step it readjusts the centroids. That is done by averaging all data points belonging to the same cluster. The average is the new cluster centroid. The process is then repeated until conversion.

The main struggle with K-means is choosing the number of clusters before initializing the algorithm. In many cases, it is impossible to know how many clusters we need. However, for this experiment we create the same number of clusters as in Ljubešić et al. (2017), which is 2000 clusters.

5. Word2Vec

In order to use words as input for K-means, they need to be represented by vectors. A very efficient model for that is Word2Vec presented by Mikolov et al. (2013). Word2Vec uses a single layer of a feed-forward neural network. For input, words are encoded as vectors with one-hot representation. Then the context window (which is determined in forehand) is observed and the hidden layer calculates weights that determine the probability of one word co-occuring with some (or more) other words. The intuition behind this is that similar words appear in similar contexts. The output is a feature matrix of words. Word2Vec is typically used to predict the next word, but in this case we use it to create the proper input for K-means clustering.

6. Implementation

6.1. Dataset

The dataset used for clustering is the slWaC v2.0, a corpus of web Slovene (Erjavec et al., 2015), which comprises of 1.2 billion tokens. It also contains lemma and morphosyntactic annotations that were not used for clustering.

For tagger training and testing, the Janes-Tag v1.2 dataset (Fišer, 2016) was used. The training portion consists of 60,367 tokens and the test portion of 7,484 tokens. The development portion was not used for this experiment. The data was tokenized and converted into one sentence per line, which was needed as input.

6.2. Clustering

The first step before clustering is converting words into vectors by Word2Vec. This is done by the Gensim library (Řehůřek and Sojka, 2010). We feed our dataset into Word2Vec continuous bag of words (CBOW) model, where we set some additional parameters. As in (Ljubešić et al., 2017), we only consider words with frequency count above 50. The default window size for English is 5, but we descrease that to 2 to capture more syntactic relation and not semantic. The other parameters take their default values[1]. We then use the Scikit-learn package (Pedregosa et al., 2011) to implement K-means clustering. We create 2000 clusters, which takes roughly 2 days.

6.3. Integration with the tagger

We use the cluster information to improve ReLDI tagger (Ljubešić et al., 2017), but in order to evaluate it correctly we also train it on CMC data. We replace the Brown clusters by K-means clusters. This takes some adaptation, because Brown clustering is hierarchical and the clusters are included together with binary paths for easier search. This is something that K-means clustering does not have because it is not hierarchical. However, the binary paths should not affect performance, only computing time, so we simply add the binary paths to K-means clusters.

The tagger is then trained on these clusters and the training portion of Janes-Tag. Training takes roughly 6 hours.

7. Results

The tagger is evaluated on the test portion of Janes-Tag in two ways: the complete morphosyntactic description (MSD) and only first two labels in description (PoS). We calculate accuracy for both these sets because that is the typical evaluation metric for classification models. Results are compared to the baseline performance (ReLDI retrained on CMC data) and to the Brown clusters model. The results are available in Table 1 below.

[1] Parameter configuration: *size=100, alpha=0.025, window=2, min_count=50, max_vocab_size=None, sample=1e-3, seed=1, workers=3, min_alpha=0.0001, sg=1, hs=0, negative=5, cbow_mean=1, hashfxn=hash, iter=5, null_word=0, trim_rule=None, sorted_vocab=1, batch_words=MAX_WORDS_IN_BATCH.*

	baseline	Brown	K-means
MSD	84.15	85.17	88.32
PoS	89.85	91.12	92.88

Table 1: Comparison of the tagger accuracy between our model (K-means) and models from Ljubešić et al. (2017).

Our model improves the baseline by 4.17% on the MSD set and 3.03% on the PoS set. It also outperforms the Brown clusters slightly. The results are unexpected, because Brown clusters had been analyzed often in research works and proved to be the most efficient. However, Brown clustering works on the intuition that similar words appear in similar contexts, and Word2Vec has that same intuition. This might be the cause of such high performance of K-means.

Especially interesting is the result for the MSD set, because high accuracy seemed hard to achieve in Ljubešić et al. (2017). Our result, although still lower than the PoS result, could be useful for morphologically rich languages that require many tags to describe morphology.

Error analysis shows that our tagger performed well in determining many tags, but might have failed in the final tag or two of the full morphological description. Even though this information is not complete, it would still provide some useful information.

In the PoS set, the most problematic category was nouns. Common nouns account for 20% of the errors, while proper nouns account for 12%. This is not unexpected, as this category is also the most diverse and might contain many words that were not seen in training. Other categories with the greatest error margin were general adverbs with 10% and general adjectives with 11%. There were many words that were annotated as elements from another language, but were mostly recognized as nouns by the tagger (10% of errors). This category is also problematic from another point of view: The words in it belong to two categories simultaneously, as they are elements from a different language, but they also belong to some word class. In these situations K-means cannot be of great help, because it is a hard-clustering technique, which means that a single word only belongs to one cluster. To further improve this, a soft clustering technique would be required.

In the MSD set, the most common errors were common nouns (masculine, singular, nominative case) and general adverbs (positive degree). These groups both account for 6% of errors.

8. Conclusions

This paper presented an implementation of K-means clustering to be used in a standard language tagger for non-standard text analysis. It outperforms the previous attempts to improve the tagger, which could be assigned to Word2Vec. K-means is an easy clustering algorithm to implement and scales well to large datasets. This speaks for the usefulness of this method. Using Word2Vec with clustering should be investigated further and combined with other clustering methods to find the most optimal one.

These results could be improved in several ways. As already mentioned, Ljubešić et al. (2017) presented several experiments where tagger was successfully improved and the final configuration is freely available. K-means together with Word2Vec could be used on that final configuration.

In this paper we created 2000 clusters for K-means, but this should be further investigated. Since there are 960 possible tag combinations for the full morphological description, it would be instightful to create exactly that many clusters. Furthermore are the tags themselves hierarchical, going from wider to more specific categories, so it would be interesting to try with some other hierarchical clustering algorihtms.

Additional improvement possibilities lie in Word2Vec parameters, where we could use a larger window size and increase of decrease the frequency of the words observed or change the default parameters. It might be useful to decrease the frequency, since non-standard language uses more diverse vocabulary and therewith less frequent words.

References

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-based *n*-gram Models of Natural Language. *Computational linguistics*, 18(4):467–479, 1992.

Jacob Eisenstein. What to do about bad language on the internet. In *Proceedings of NAACL-HLT*, pages 359–369, 2013.

Tomaž Erjavec, Nikola Ljubešić, and Nataša Logar. The slWaC Corpus of the Slovene Web. *Informatica*, 39(1):35, 2015.

Darja Fišer. Gold-Standard Datasets for Annotation of Slovene Computer-Mediated Communication. *The 10th Workshop on Recent Advances in Slavonic Natural Languages Processing*, *RASLAN 2016*, page 29, 2016.

Dekang Lin and Xiaoyun Wu. Phrase Clustering for Discriminative Learning. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural*

Language Processing of the AFNLP: Volume 2 - Volume 2, pages 1030–1038. Association for Computational Linguistics, 2009.

Nikola Ljubešić, Tomaž Erjavec, and Darja Fišer. Adapting a State-of-the-Art Tagger for South Slavic Languages to Non-Standard Text. *In Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, pages 60–68, 2017.

James MacQueen et al. Some Methods for Classification and Analysis of Multivariate Observations. *In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability,* pages 281-297, 1967.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, an Noah A Smith. Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 380-390, 2013.

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

Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, 2010.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word Representations: A Simple and General Method for Semi-Supervised Learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.