Evaluation of vector embedding models in clustering of text documents

Tomasz Walkowiak
University of Science and Technology,
Faculty of Electronics,
Wybrzeze Wyspianskiego 27,
Wroclaw 50-370, Poland
tomasz.walkowiak@pwr.wroc.pl

Mateusz Gniewkowski
University of Science and Technology,
Faculty of Computer Science and Management,
Wybrzeze Wyspianskiego 27,
Wroclaw 50-370, Poland
mateusz.gniewkowski@pwr.wroc.pl

Abstract

The paper presents an evaluation of word embedding models in clustering of texts in the Polish language. Authors verified six different embedding models, starting from widely used word2vec, across fastText with character n-grams embedding, to deep learning-based ELMo and BERT. Moreover, four standardisation methods, three distance measures and four clustering methods were evaluated. The analysis was performed on two corpora of texts in Polish classified into subjects. The Adjusted Mutual Information (AMI) metric was used to verify the quality of clustering results. The performed experiments show that Skipgram models with n-grams character embedding, built on KGR10 corpus and provided by Clarin-PL, outperforms other publicly available models for Polish. Moreover, presented results suggest that Yeo–Johnson transformation for document vectors standardisation and Agglomerative Clustering with a cosine distance should be used for grouping of text documents.

1 Introduction

A number of digital repositories of texts enlarge each year. The variety of tools for natural language processing and the quality of their performance are growing steadily. That opens possibilities for automatic categorisation of text documents in terms of the subject areas in any digital collection of documents. It is also an important problem for researchers from different areas of the humanities and social science (Eder et al., 2017).

Commonly used methods rely on representing documents with feature vectors and using clustering algorithm (Hastie et al., 2009) to assign documents to some groups. The classical feature vectors are based on the bag-of-words technique (Harris, 1954). Components of these vectors represent frequencies (weighted) of occurrences of words/terms in individual documents. The contemporary state-of-the-art technique is word2vec (Le and Mikolov, 2014), where individual words are represented by high-dimensional feature vectors trained on large text corpora. This technique is constantly being improved. This is demonstrated by the most recent propositions of algorithms like ELMo (Peters et al., 2018) or BERT (Devlin et al., 2018). Choosing the most useful clustering algorithm is not a trivial task since there is a large number of them. Just to mention the most popular ones like K-means (K.Jain, 2009), Agglomerative Hierarchical Clustering (Day and Edelsbrunner, 1984) and Spectral Clustering (Ng et al., 2002). Moreover, the results of clustering are strongly dependent on the chosen distance measure and the used method of an input data standardisation. The entire workflow, described above, expresses many factors which can influence results of the text exploration. It is difficult to control them and thus, might lead to unpredictable outcomes of the experiment. It becomes challenging for texts in an inflected language such as Polish.

The main aim of this research is the evaluation of clustering accuracy on documents in Polish, using publicly available word embedding models. We conducted our experiments to answer the following research questions: What is the best method (i.e. the word2vec model, standardisation method, distance metric and clustering algorithm) for subject grouping texts in Polish? The experiments were performed on two corpora with texts assigned to subject groups. We analysed the quality of results (defined by AMI metric (Romano et al., 2016)) in a function of method options.

Proceedings of Recent Advances in Natural Language Processing, pages 1304–1311,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_149

Some works studied the quality of word2vec models for Polish (Piasecki et al., 2018; Mykowiecka et al., 2017; Kocon and Gawor, 2019), but they focus on single words, not on an application of the word embeddings to represent whole documents for a clustering purpose.

The paper is structured as follows. In Section 2, we describe in details word embedding techniques and list the models examined in the work. Next, we provide technicalities about the methods we compared and finally in Section 4 we present test corpora used in the study as well as results of the comparative study.

2 Word Embeddings

2.1 Techniques of Word Embedding

Word embedding is an approach of text analysis based on the assumption that individual words can be represented by high-dimensional feature vectors. It is based on the hypothesis that relationships (distances) between vector representations of words can be related to semantic similarities of words. The models are built on large text corpora by observing co-occurrence of words in similar contexts. One of the most popular technique, *word2vec*, is based on neural networks (Le and Mikolov, 2014). The authors proposed two approaches: CBOW and Skipgram. In the first one, the aim is to predict a word based on context words. The Skipgram model does the opposite task, it predicts context words from a given word. In the classical word2vec (Le and Mikolov, 2014) technique each word (form from the text) is represented by a distinct vector, which might be a problem for a language with large vocabularies and rich inflexion like Polish is. The first solution to this problem was to build models based on word lemmas (Mykowiecka et al., 2017), however, such a technique requires a morphological analysis of all texts in a training corpus. Next, in (Bojanowski et al., 2017) authors extend the Skipgram model by building a vector representation of character n-grams and constructing the word representation as the sum of the character n-grams embeddings (for n-grams appearing in the word). It could decree the model size and allows to generate word embedding for words not seen in a training corpus.

The next step forward was introducing (Grave et al., 2018) the extension of the original CBOW model (Le and Mikolov, 2014) with position weights and subword information (character n-grams).

The newest approaches are inspired by deep-learning algorithms. In a recently introduced ELMo (Peters et al., 2018), word embeddings are defined by the internal states of a deep bidirectional LTSM language model (biLSTM), which is trained on a large text corpus. What is important, ELMo looks at the whole sentence before assigning an embedding to each word in it. Therefore, the embeddings are sentence aware and could solve a problem of polysemous words (words with multiple meanings). Another approach similar to ELMo is BERT (Devlin et al., 2018). It is also a bidirectional representation, but it is jointly built on both the left and the right context. The available BERT models[1] are multilingual and pre-trained on two unsupervised tasks: masked language modelling and next sentence prediction.

Word embeddings can be simply used to generate feature vectors for document clustering by averaging vector representations of individual words occurring in a document. This approach is known as *doc2vec* and used for example in fastText (Joulin et al., 2017) algorithm. In the case of BERT, we get sentence embeddings, but the approach used for word models can be repeated here as well (as an average of sentence embeddings).

2.2 Available Models for Polish

There are two groups working on publicly available word embedding models for Polish: IPI PAN[2] and Clarin-PL[3]. The IPI PAN provides[4] a set of more than 100 CBOW and Skipgram models generated from data consisting of National Corpus of Polish (NKJP) (Przepiorkowski et al., 2012) and Wikipedia (Wiki). Some of them are generated only for lemmas, others of words from texts (forms). For tests we have selected the Skipgram model (i.e. *nkjp+wiki-forms-all-300-skipg-hs-50*). The model was generated by gensim tool[5]. It assigns a distinct vector to each word.

The Clarin-PL provides[6] 16 models generated by fastText software[7] on larger than in a previous case corpus (Kocon and Gawor, 2019). They are joint models of words and character n-grams

[1] https://github.com/google-research/bert

[2] https://ipipan.waw.pl/en/

[3] https://clarin-pl.eu/en/home-page/

[4] http://dsmodels.nlp.ipipan.waw.pl

[5] https://radimrehurek.com/gensim/

[6] http://hdl.handle.net/11321/606

[7] https://fasttext.cc/

able to produce vectors for unknown words (Bojanowski et al., 2017). For tests, we have selected two Skipgram models based on forms (*KGR10*) and lemmas (*KGR10_lemma*). The second group of sources of word2vec models for Polish are web pages of word embedding tools like fastText, ELMo and BERT. They were trained on Polish Common Crawl and Wikipedia. However, the BERT model was trained on many languages in parallel. The details on used models are summarised in Table 1.

3 Methods

3.1 Metrics and Clustering

A distance measure needs to distinguish contrasting samples. Sometimes that contrast is well defined, and we know what kind of behaviour we require from the chosen function. In natural language processing commonly used function is a cosine distance as i.e. it does not distinguish documents, described as a vector of most frequently occurred words in the corpus, that have a linear dependence between features. It also works well with sparse high-dimensional space and is less *noisy* than euclidean (Kriegel H-P., 2012). Models we want to compare have different properties, so relevant distance function is even less obvious. In our work we decided to focus on *cosine*, *Bray–Curtis* and *Euclidean distance*.

We also tried few different variations of them but the results were not significantly different.

In order to group data, we decided to use following methods (mind that two of them use only Euclidean distance or L_k norm in general, because otherwise algorithms may stop converging).

1. *K-means* (K.Jain, 2009) algorithm is a classic method that assigns labels to the data, basing on a distance to the nearest centroid. Centroids are moved iteratively until all clusters stabilise.

2. *Agglomerative hierarchical clustering (AC)* (Day and Edelsbrunner, 1984) is a method that iteratively joins subgroups basing on a *linkage criterion*. In this paper, we present result for the average linkage clustering.

3. *Spectral clustering (SC)* (Ng et al., 2002) is based on the Laplacian matrix of the similarity graph and its eigenvectors. The least significant eigenvectors create new, lower dimensional space that is used with a *K-means* algorithm.

4. *Expectation–maximisation (EM)* (Bilmes et al., 1998) is used to estimate parameters in statistical models, Gaussian mixture model in our example. Gaussian mixture model assumes that data is generated from a finite number of Gaussian distributions.

3.2 Standardisation

Standardisation of input data is usually necessary, because many machine learning algorithms requires features to have a normal distribution and, probably more important in clustering algorithms, a similar scale. In our work, we decided to use following methods:

Table 2: Symbols description

X	feature vector (column of the input matrix)
$\overline{X}$	mean value of the feature
X_{min}, X_{max}	minimal/maximal value of the feature
$x_i, \hat{x}_i$	new/old value of the feature of i-th sample
σ	standard deviation of the feature
λ	power parameter that is estimated through maximum likelihood

1. Min–Max scaling - the most popular way of scaling data. Returned values are in range from 0 to 1:

$$\hat{x}_i = \frac{x_i - X_{min}}{X_{max} - X_{min}}$$

2. Z-score normalisation - one of the classic method of standardisation. It results in a distribution with a standard deviation equal to 1:

$$\hat{x}_i = \frac{x_i - \overline{X}}{\sigma}$$

3. Yeo–Johnson transformation - a member of power transform functions that allows negative values of input (Yeo and Johnson, 2000):

$$\hat{x}_i = \begin{cases} \frac{(x_i+1)^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0, x \geq 0 \\ \log(x_i + 1), & \text{if } \lambda = 0, x \geq 0 \\ -\frac{(-x_i+1)^(2-\lambda)-1}{2-\lambda}, & \text{if } \lambda \neq 2, x < 0 \\ -\log(-x_i + 1), & \text{if } \lambda = 2, x < 0 \end{cases}$$

[8] CBOW with position weights

Table 1: Used embedding models

name	method	feature	tool	size	address
IPIPAN	CBOW	forms	gensim	300	dsmodels.nlp.ipipan.waw.pl
KGR10	Skipgram	forms, character n-grams	fastText	300	hdl.handle.net/11321/606
KGR10_lemma	Skipgram	lemmas,character n-grams	fastText	300	hhdl.handle.net/11321/606
fastText	CBOW+[8]	orths, character 5-grams	fastText	300	fasttext.cc
elmo	ELMo	forms	ELMo	1024	vectors.nlpl.eu/repository/11/167.zip
bert	BERT	multilingual forms, sentences	BERT	768	github.com/google-research/bert

3.3 Quality Metrics

Evaluation of clustering quality may be performed in two different ways: with external knowledge of sample membership or without it. The first way is usually better if we have already labelled data and using supervised learning is none of our options. For example, we know that the clustering problem we want to solve concerns similar data we have labelled, which is a case in this work. We compare how different vector representation of documents, which have an assigned label to it, can be clustered.

There are plenty of clustering quality measures that have a different interpretations, like *purity*, *V-measure* or *Rand Index* (Amigo et al., 2009). In our work, we decided to use corrected for a chance measures where the result does not increase with several clusters for randomly chosen labels. Two most common metrics are *Adjusted Rand Index* (ARI) and *Adjusted Mutual Information* (AMI) (Vinh et al., 2010). According to (Romano et al., 2016) AMI is better suited to our problem as documents types are often unbalanced. It gives more weight to a clustering solutions with purer small groups than to minor mistakes in bigger ones.

Adjusted mutual information score is one of the information theoretically based measures. It is based on mutual information (MI) which comes naturally from entropy.

Table 3: Symbols description

X, Y	set of classes/clusters
H	Entropy
MI	mutual information
NMI	normalized mutual information
AMI	adjusted mutual information
x_i, y_i	i-th element of X/Y (class or cluster)
$P(x_i), P(y_i)$	probability of the document being in i-th class or cluster
$P(x_i \hat{y}_j)$	intersection of $P(x_i)$ and $P(y_j)$
$E(MI)$	expected value of MI

$$H(X) = \sum_i P(x_i) \log \frac{1}{P(x_i)}$$

$$MI(X,Y) = \sum_i \sum_j P(x_i \cap y_j) \log \frac{P(x_i \cap y_j)}{P(x_i)P(y_i)}$$

The problem with mutual information is that the maximum is reached not only when labels from one set (clusters) matches perfectly those from the other (classes), but also when they are further subdivided. The simple solution for that is to normalise MI by mean of entropy of X and Y:

$$NMI(X,Y) = \frac{MI(X,Y)}{(H(X) + H(Y))/2}$$

Normalised mutual information can be further improved by subtracting expected value of MI from nominator and denominator:

$$AMI(X,Y) = \frac{MI(X,Y) - E(MI)}{(H(X) + H(Y)/2 - E(MI)}$$

This is what is called ”corrected for a chance”. The general form of AMI was proposed in (Hubert and Arabie, 1985).

4 Experiments

4.1 Data Sets

We performed tests on two collections of text documents in Polish: $Press$ and $Rewievs$. The first corpus ($Press$) comprises Polish press news. It is a complete, high quality and well defined data set. The texts were assigned by press agency to five subject categories. All the subject groups are well separable from each other and each group contains a reasonably large number of members. In the study (Walkowiak and Malak, 2018), the authors reported 95.5% accuracy achieved on this data set in supervised classification. There are ca. $6,500$ documents in total in this corpus.

The second corpus (*Reviews*) consists of reviews of scientific works from 21 different science areas. The achieved accuracy on this data set by fastText (Joulin et al., 2017) in supervised classification was 90.7% (after division 2:1 for training and testing). There are ca. $10,500$ documents in this corpus.

4.2 Idea

The goal of our experiments was to find the best performing word embedding model in a clustering problem. In order to do that, first, we checked how standardisation affects results and picked one of the methods to use it in further tests. Then we compared several models using different clustering approaches with and without standardisation. In order to generate feature vectors for documents (doc2vec) we averaged word/sentence embeddings for every text in the dataset.

4.3 Choosing the Standardisation Method

We performed our first experiment as follows: having the documents $d_i \in D$ represented as doc2vec vectors from KGR10_lemma model, we performed several tests to evaluate the quality measure (AMI) of multiple clustering algorithms with different distance functions. The results are given in the Figure 1 and Figure 2. It can be observed that for EM, K-means and SC standardisation does not significantly improve the results. What is more, for Euclidean distance, data scaling may blur the distances between points and worsen the quality of the solution. On the other hand, usage of standardisation methods with Agglomerative Clustering (AC) algorithm improves obtained results. It is not surprising as the linkage method strongly depends on variance especially when using a cosine distance. On average (the average height of the AMI score) the best method of standardisation turned out to be Yeo–Johnson transformation, so we used it in subsequent experiments.

4.4 Model Comparison

In order to compare how the chosen model affects quality, we performed multiple tests, similar to the previously conducted. They were evaluating the quality measure due to used doc2vec representations, generated from models described in Section 2. The results of clusterisation of the original data can be observed in figures 3 and 5 alongside with the results of standardised vectors with Yeo–Johnson transformation in figures 4 and 6. It can be noticed that standardisation has minor influence on Spectral Clustering (SC). It either slightly improves or does not deteriorate results. The only exception is Euclidean distance where standardisation can blur and therefore worsen the score. It is clearly visible for Agglomerative Clustering (AC) which, after standardisation and with a cosine or a Bray–Courtis distance, works best. Using standardised version of K-means algorithm with any of proposed models is rather not suitable since standardisation does not necessarily increase the score, however this classic approach is usable with the given problem, especially that it strongly depends on centroids which usually are quite useful. We expected that standardisation will have a positive influence on Gaussian mixture model (EM) which assumes data is generated from some number of Gaussian distributions and standardisation should make that data more Gaussian-like. It is probably the case with those models that have higher score in figures 4 and 6 than in figures 3 and 5, but it is not a rule.

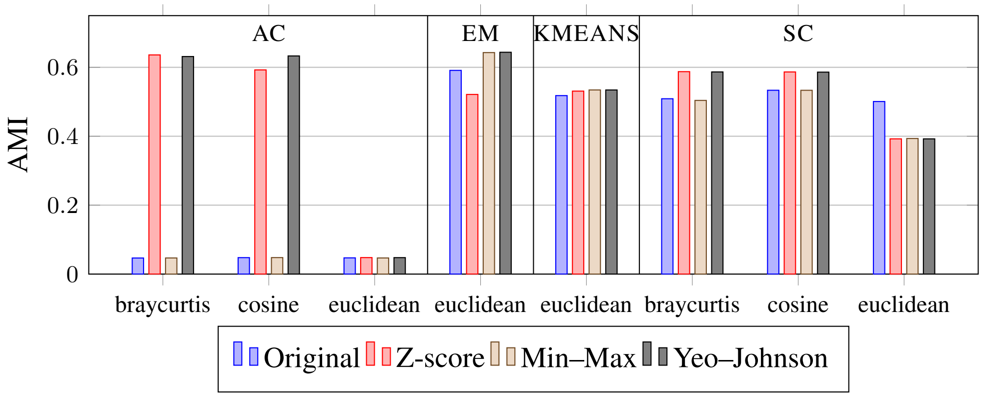

Figure 1: Standarization - PRESS

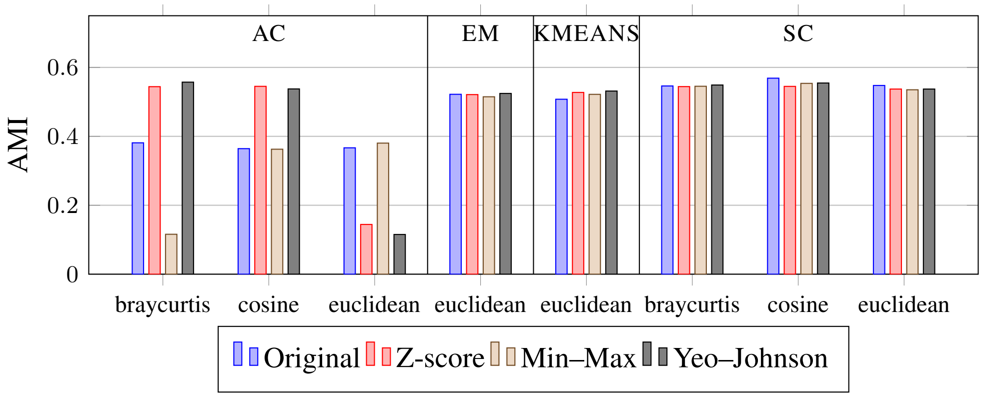

Figure 2: Standarization - REVIEWS

5 Conclusion

As a conclusion, we would like to recommend using Yeo–Johnson transformation to standardise doc2vec embedding and if a task is to group documents, Agglomerative Clustering (AC) with a cosine distance. For researchers who deal with Polish datasets, we strongly recommend using KGR10_lemma or KGR10 word2vec models (Kocon and Gawor, 2019)[9]. First of them gives better results but the second one (only slightly worse) is much faster in a usage, since it does not require a time consuming lemmatization of texts. We are now working on implementing the best selected workflow as a part of WebSty (Eder et al., 2017), an online tool[10] aimed for researchers in humanities and social science working with texts in Polish.

Although, that ELMo and BERT perform great in many tasks in NLP our results show otherwise. Two factors can be responsible for this. First, we used already trained models downloaded from the addresses shown in Table. 1. As they might not be the best quality for the Polish language, we currently training our own model to verify this hypothesis. The second reason may be that both BERT and ELMo do not work well with the discussed problem. It is hard to find any article dealing with document clustering problem using those methods.

We plan to test other methods of composing document vectors, i.e. representing documents by several concatenated vectors. We want to test two approaches. One is based on a division of documents into parts and generating doc2vec for each part. And the second, based on clustering of word embeddings into a predefined number of groups and using centroids as elements of final document vectors.

The work was funded by the Polish Ministry of Science and Higher Education within CLARIN-PL Research Infrastructure.

[9] http://hdl.handle.net/11321/606

[10] http://ws.clarin-pl.eu/websty.shtml

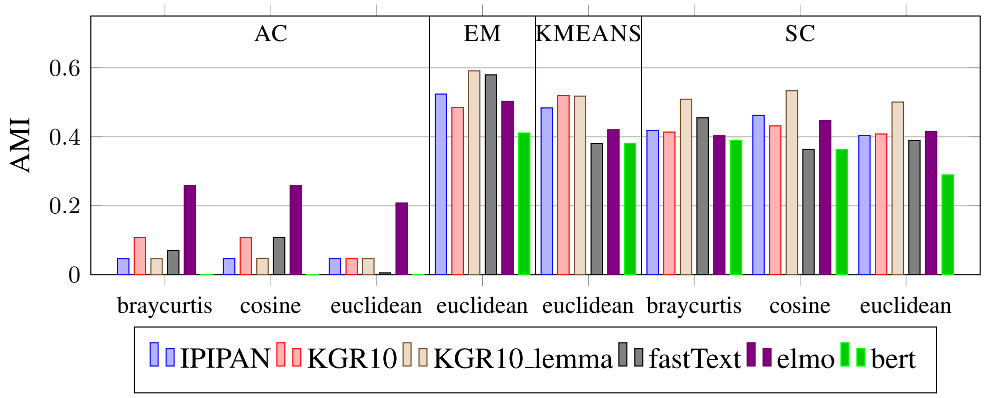

Figure 3: Model Comparison - PRESS

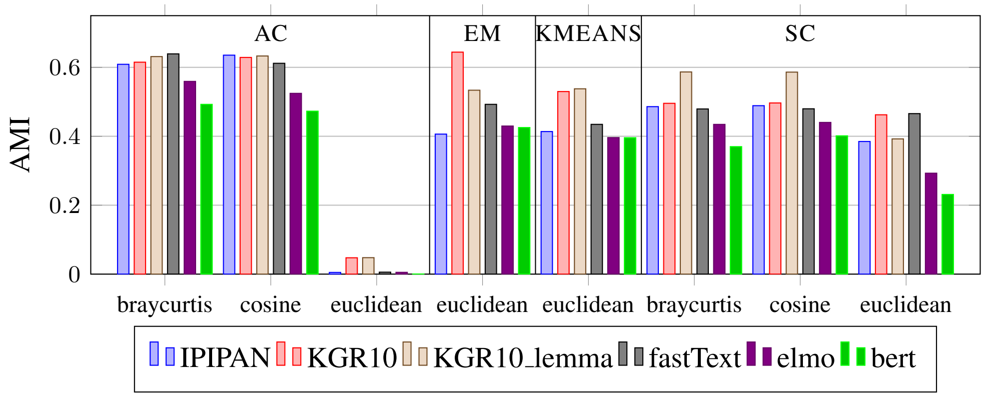

Figure 4: Model Comparison - PRESS (standardised with Yeo–Johnson transformation)

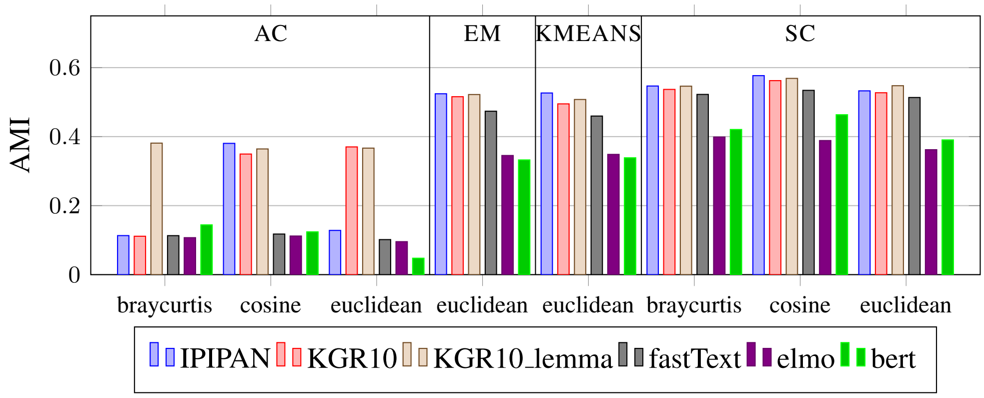

Figure 5: Model Comparison - REVIEWS

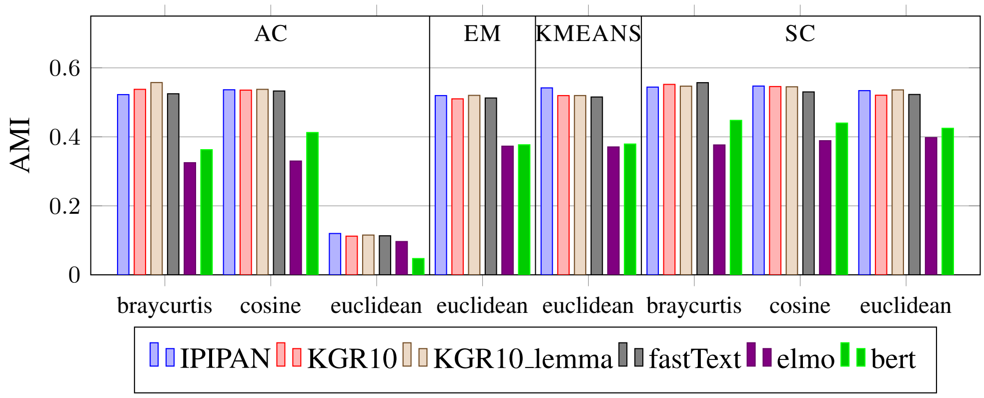

Figure 6: Model Comparison - REVIEWS (standardised with Yeo–Johnson transformation)

References

Enrique Amigo, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval* 12(4):461–486.

Jeff A Bilmes et al. 1998. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute* 4(510):126.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146.

William H. E. Day and Herbert Edelsbrunner. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification* 1(1):7–24. https://doi.org/10.1007/BF01890115.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Maciej Eder, Maciej Piasecki, and Tomasz Walkowiak. 2017. An open stylometric system based on multilevel text analysis. *Cognitive Studies — Etudes cognitives* 17. https://doi.org/10.11649/cs.1430.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Zellig S Harris. 1954. Distributional structure. *Word* 10(2-3):146–162.

Trevor J. Hastie, Robert John Tibshirani, and Jerome H. Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*. Springer series in statistics. Springer, New York. Autres impressions : 2011 (corr.), 2013 (7e corr.).

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification* 2(1):193–218. https://doi.org/10.1007/BF01908075.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, pages 427–431. http://aclweb.org/anthology/E17-2068.

Anil K.Jain. 2009. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters, Volume 31, Issue 8,* pages 651–666. https://doi.org/10.1016/j.patrec.2009.09.011.

Jan Kocon and Michal Gawor. 2019. Evaluating KGR10 polish word embeddings in the recognition of temporal expressions using bilstm-crf. *CoRR* abs/1904.04055. http://arxiv.org/abs/1904.04055.

Schubert E. Zimek A. Kriegel H-P. 2012. A survey on unsupervised outlier detection. *Statistical Analysis and Data Mining* pages 363–387.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. pages 1188–1196.

A. Mykowiecka, M. Marciniak, and P. Rychlik. 2017. Testing word embeddings for polish. *Cognitive Studies — Etudes cognitives* 17. https://doi.org/10.11649/cs.1468.

Andrew Y Ng, Michael I Jordan, and Yair Weiss. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*. pages 849–856.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Maciej Piasecki, Gabriela Czachor, Arkadiusz Janz, Dominik Kaszewski, and Pawel Kedzia. 2018. Wordnet-based evaluation of large distributional models for polish. In *Proceedings of the 9th Global Wordnet Conference (GWC 2018)*. Global WordNet Association.

A. Przepiorkowski, M. Banko, R. L. Gorski, and B. Lewandowska-Tomaszczyk, editors. 2012. *Narodowy Korpus Jzyka Polskiego*. Wydawnictwo Naukowe PWN, Warszawa.

Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. 2016. Adjusting for chance clustering comparison measures. *The Journal of Machine Learning Research* 17(1):4635–4666.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research* 11(Oct):2837–2854.

Tomasz Walkowiak and Piotr Malak. 2018. Polish texts topic classification evaluation. In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*. INSTICC, SciTePress, pages 515–522. https://doi.org/10.5220/0006601605150522.

In-Kwon Yeo and Richard A Johnson. 2000. A new family of power transformations to improve normality or symmetry. *Biometrika* 87(4):954–959.