Content-based recommendations in e-commerce services

John Smith[*]

Warsaw University of Technology
Faculty of Mathematics and Information Science
ul. Koszykowa 75
00-662 Warsaw, Poland
john@smith.com

February 23, 2018

Abstract

Recommendation systems play an important role in modern e-commerce services. The more relevant items are presented to the user, the more likely s/he is to stay on a website and eventually make a transaction. In this paper, we adapt some state-of-the-art methods for determining similarities between text documents to content-based recommendations problem. The goal is to investigate a possibility of improving an existing recommendation system being a part of Allegro e-commerce platform using semantic text analysis methods. As a conclusion we show, that there is no significant difference between examined methods and previously used elasticsearch-based query in content-based recommendation task.

Keywords: recommendations, natural language processing, word embedding, semantics, allegro

I. Introduction

Recommendation systems are commonly used in e-commerce services. Such system give profit to both a user as well as a website owner. They allow the user to get to an information s/he could not find, or even know, that such information exists. They also attract users to a service making them more likely to buy something, increasing website company's profit thereby. The key issue of recommendations generation is how suggested items are relevant to these which the user is interested in.

Generally we can divide recommendation systems into two groups: collaborative and content-based filtering. The first one assumes that user is likely to be interested in items which also users similar to s/he were interested in. In this paper we are focusing on the second group in which recommended items are similar to these that the user liked so far.

A detailed problem comes from the on-line auction site Allegro. Allegro — the biggest marketplace platform in Eastern Europe — contains a section presenting text articles concerning products available via the platform. Next to article currently displayed by a user, there is a list of links to articles similar to given one. Previous recommendations determining mechanism is based on elasticsearch query and uses both an article text and some metadata attached to the article.

In this paper we strive to check if some semantic text analysis methods, including newly proposed word embeddings, are able to replace previously used Allegro method. Examined methods are topic modeling methods: latent semantic analysis [2], latent Dirichlet allocation [1] and word embeddings methods: word2vec [5], fastText [3] and GloVe [7].

In section 2 we make an analysis of the articles dataset and also we describe data pre-

[*] A thank you or further information

processing we performed. Section 3 is about methods used for building the model and next evaluate results. In section 4 we present the results and try to interpret them. The last section concludes the papers and proposes a direction of future work.

II. Dataset

As mentioned our dataset comes form Allegro e-commerce platform. It consists of 20000 text articles describing several categories of products available via Allegro platform. Single record consists of article contents and metadata attached by an author of the article. As significant metadata we consider „category" and „keywords".

All articles are written in Polish. The vocabulary based on the articles set contains many industry-specific words like brands and models names, books titles and technical words. Moreover raw articles contain some tags responsible for images and hyperlinks present on the website.

A standard activity before building a model is preprocessing of a raw dataset. The following enumeration describes steps of the preprocessing that we performed.

1. Cleansing the text from the redundant, previously mentioned tags. From the viewpoint of semantic analysis they are useless or even noxious. That is why we remove them using properly constructed regular expressions. An example of such a tag is `[2_new.jpg] (http: // (...)'2_new.jpg')` placing the picture in the middle of the text (the content of the URL removed due to confidentiality reasons).

2. Removal of stopwords — usually short words carrying very little information about the actual document contents, e.g. „in", „from", „because" etc. Removing them reduces the number of words in a document and the processing time thereby.

3. Converting all words of a document to lowercase letters. It helps to unify words with the same meaning, but written using both uppercase and lowercase letters.

4. Breaking down the words connected with a hyphen. Experience at a later lemmatisation stage shows that the tool used to it (Morfologik [6]) does not cope with these types of words and leaves them in the unchanged grammatical form. This made it necessary to build our own tool that breaks such words into sub-words compatible with the lemmatiser.

5. Tokenization and lemmatisation. This is the most important element of the process. It's about separating text into individual words and converting the words with the same meaning, and different grammatical form to the same form. Complexity of the Polish language and the number of exceptions that this language has are making this problem harder than in many other languages. To carry out this operation, we used the Morfologik tool [6].

The above steps lead the data to a state in which we can apply examined semantic techniques of text analysis. The dictionary built on the preprocessed corpus contains 98174 unique words and 7409145 all words (with repetitions). The overwhelming majority of the words of the dictionary built on the body are the words appearing rarely.

III. Methods

Emerging popularity of word embeddings methods started

Next we had to choose an evaluation method....

i. Model building

Below enumeration contains text analysis methods we decided to adapt for our content-based recommendations task.

1. Latent semantic analysis

2. Latent Dirichlet allocation

3. Word2Vec
4. GloVe
5. FastText

Hyperparameters of the methods...

Word2Vec authors offer pretrained model but only for English language. For Polish it's more difficult... Lack of presumably important words like brand and model names etc.

Text requiring further explanation[1].

ii. Evaluation

In order to compare the methods used to determine the similarity between the articles necessary there is the formalization of certain measures of this similarity.

In practice, however, it is rarely worth the value of a given element of the ranking is adequate to the query generating this ranking.

Evaluation of recommendations generation methods is a nontrivial task. A degree of similarity of two articles can be perceived differently by different persons. Therefore we decided to evaluate methods' performance with .... First with each tested method we choose 6 most similar articles for 50 randomly chosen base articles. Next we split them into pairs: base article — similar article; it gave us $5 * 50 = 250$ pairs for a method. After that 5 persons individually evaluated a similarity of pairs of articles giving scores in scale 1-10. Finally we took an average score for each pair and it allowed us to calculate an average score for each method. Besides tested methods we evaluated also previous Allegro method and randomly generated pairs of articles just for a comparison purpose.

IV. Results and discussion

In order to compare aforementioned methods we performed described expert evaluation. The evaluated methods were: LSA, LDA, word2vec, GloVe, FastText, previous Allegro elasticsearch-based method and randomly chosen pairs of articles.

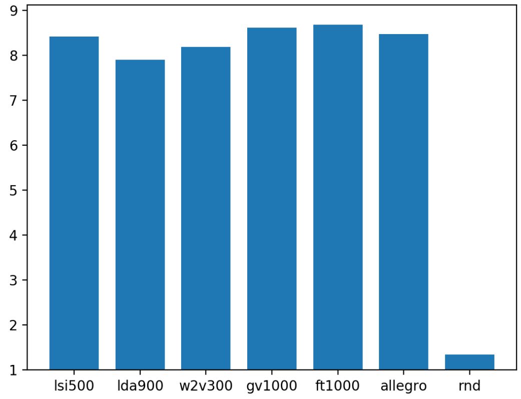

Figure 1

Expert evaluation suggests that none of the tested methods is significantly different from others. The difference between the best method (FastText) and the worst one (LDA) is around 9%. Each of the tested methods also gave a significantly better result than the random method.

To answer the question posed at the beginning of this work, i.e. whether the examined text analysis methods adapted to content-based recommendation task are able to match the previous method used in Allegro, we perform a test that's goal is to check whether there are grounds to believe that the results of any of the methods is statistically significantly different from the rest of the examined methods.

The statistical test that we carry out is the Kruskal-Wallis test. In the test the null hypothesis H_0 assumes an equality of distributions in populations from which samples originated. As input data the Kruskal-Wallis test takes samples corresponding to expert evaluation of each method consisting of an average score made by users for the most relevant recommendation for each of the examined base articles.

In the test, the level of significance is $\alpha = 0.05$. Finally, as the result of the Kruskal-Wallis test we received $p = 0.0571 > \alpha$, on the basis of which we state that there are no reasons to reject the null hypothesis — the quality differ-

[1] Example footnote

ence between of recommendations generated by the tested methods is not statistically significant. It means that neither of the tested semantic methods sticks out from others, nor the previous Allegro method is significantly better/worse than other tested methods.

V. Conclusion

After analyzing the test results we can answer: yes, each of the examined methods of natural language semantic analysis in their best configuration can be adapted to generate article content-based recommendations. This means we can replace previously used method with e.g. FastText and any user should not feel any difference in recommendations quality.

Using methods of semantic text analysis allows to capture the hidden similarities between documents, where the documents combine not the same words or synonyms, but some abstract concepts related to each other. An important advantage of semantic methods compared to the previous one is the fact that they are based only on article contents. This frees the authors of the articles from providing additional metadata, which the previous method uses in large extent.

References

[1] D. M. Blei, A. Y. Ng, M. I. Jordan, *Latent Dirichlet Allocation*, Journal of Machine Learning Research, tom 3 num. 4–5, 2003

[2] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, *Indexing by latent semantic analysis*, Journal of the American Society for Information Science, tom 41, num. 6, 1990

[3] A. Joulin, E. Grave, P. Bojanowski T. Mikolov, *Bag of Tricks for Efficient Text Classification*, Facebook AI Research, 2016

[4] M. J. Kusner, Y. Sun, N. I. Kolkin, K. Q. Weinberger, *From Word Embeddings To Document Distances*, International Conference on Machine Learning (ICML), 2015

[5] T. Mikolov, K. Chen, G. Corrado, J. Dean, *Efficient Estimation of Word Representations in Vector Space*, International Conference on Machine Learning (ICML), 2013

[6] Morfologik `http://morfologik.blogspot.com/` (dostęp 07.05.2017)

[7] J. Pennington, R. Socher, C. D. Manning, *GloVe: Global Vectors for Word Representation*, Computer Science Department, Stanford University, Stanford, CA 94305, 2014