АВТОМАТИЧЕСКОЕ ИЗВЛЕЧЕНИЕ ПАРАМЕТРОВ ПРОДУКТОВ ИЗ ТЕКСТОВ ОТЗЫВОВ ПРИ ПОМОЩИ ИНТЕРНЕТ-СТАТИСТИК

Марчук А. А. (aamarchuk@gmail.com)
Санкт-Петербургский государственный университет, Санкт-Петербург, Россия

Уланов А. В. (alexander.ulanov@hp.com)
Научно-исследовательская лаборатория Хьюлетт-Паккард в России

Макеев И. В. (ilya.makeev@gmail.com)
Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Санкт-Петербург, Россия

Чугреев А. А. (artemij.chugreev@gmail.com)
Санкт-Петербургский государственный политехнический университет, Санкт-Петербург, Россия

Ключевые слова: анализ мнений, извлечение информации, параметры продуктов, классификация

EXTRACTING PRODUCT FEATURES FROM REVIEWS WITH THE USE OF INTERNET STATISTICS

Marchuk A. A. (aamarchuk@gmail.com)
St. Petersburg State University, St. Petersburg, Russia

Ulanov A. V. (alexander.ulanov@hp.com)
Hewlett-Packard Labs Russia

Makeev I. V. (ilya.makeev@gmail.com)
St. Petersburg State University of Information Technologies, Mechanics and Optics, St. Petersburg, Russia

Chugreev A. A. (artemij.chugreev@gmail.com)
St. Petersburg Polytechnical University, St. Petersburg, Russia

The paper studies the task of extracting product features from reviews. We consider this task as a classification problem and propose a number of classification features. These features are computed using different statistics returned by queries to Yandex search engine, the Internet library and the Russian National Corpus. To justify our approach, we create and manually label a product features dataset, compute the proposed classification features and conduct classification experiments. The results produced by various classifiers applied to different subsets of the data show the feasibility of our approach. We also look at the usefulness of the proposed classification features.

Keywords: opinion mining, sentiment analysis, information extraction, product features, classification

1. Introduction

A lot of useful information is stored in user-generated content, especially when it contains opinions. These days, users are able to express their opinions and write reviews about almost everything on the Web. Opinion mining or sentiment analysis area of study analyzes such kind of content. Its ultimate goal is to detect opinionated texts and extract who and when expressed which degree of positivity towards which entity or its attribute [12]. Then such tuples can be analyzed computationally. In this work, we are focusing on the problem of entity extraction, or, more specifically, mining product features from reviews.

The task of mining product features can be considered as information extraction task [15], or in particular, relationship extraction problem, when one mines relationships for a given product. Many methods from those fields were adapted to the problem of mining product features. One of the first works [9] dealing with this problem suggests that most frequent nouns and noun phrases in reviews are product features. Infrequent features are extracted by relationships with the same opinion words that accompany frequent features. Paper [2] proposes several useful features to detect noun phrases as product features. Pointwise mutual information (PMI) is computed between a candidate phrase and a product with a relationship discriminator. An example of the latter is “scanner comes with”, where “scanner” is a product, and “comes with” is a discriminator. PMI is also computed between a product and a candidate noun phrase. Statistics for PMI is gathered from a Web search. Other features used in [2] are WordNet’s component/parts relationships. The authors of [7] deal both with explicit and implicit product feature extraction. They perform classification into feature groups as well. Dependency parsing is employed in [17]. Hidden Markov models (HMM) are used and part of speech information is employed in [11]. Conditional random field (CRF) classifier is used in [10]. Token, part of speech, dependency path, word distance and opinion class of a sentence are used as classification attributes there. Another line of work is concerned with the use of topic modeling. Multi-grained topic model is proposed in [16], however, opinion words and product features are not distinguished into separate groups. The authors of [3] construct a localized Latent Dirichlet Allocation (LDA) model that allows them to perform clustering of product aspects and to infer sentiment orientation of them.

Most works on sentiment analysis for Russian are devoted to either sentiment classification or opinion word mining, for example [5], [14] and [4]. The latter approaches the task of opinion lexicon generation as a classification task.

This paper is motivated by development of a service for automatic sentiment analysis [6] and addresses the problem of product features extraction in Russian. We consider this problem as a classification task. We build a labeled dataset of product features extracted from product reviews, propose a number of attributes and use them to perform supervised classification. We use attributes proposed in several other works [2], [1], as well as those motivated by common sense. We also study the usefulness of features and the performance of various classifiers.

2. Classification features

We consider the problem of product feature extraction as a classification task. Candidates are extracted from text and classified into two classes: feature and not feature. Candidates can be words or phrases. The following classification features are employed: frequency, opinion word proximity, weirdness, TF-IDF, PMI. Below are their definitions. Section 3 contains a description of their implementations.

Frequency

Frequency is computed with the following formula:

$$freq_{corpus}\ (c) = \frac{N(c)}{N},$$

where *N(c)* is the number of occurrences of the candidate *c* in the corpus of size *N*.*N(c)* and *N* may be words, phrases or documents. Further, we will compute word frequency and document frequency.

Opinion word proximity

An opinion word lexicon is needed to compute this feature. The trivia is that if there is an opinion word near the candidate, then it is probable that the opinion is expressed about it and it may be a product feature. We compute the number of documents in which the opinion word *ow* is in proximity of *p* words within the candidate *c*.

Weirdness

Weirdness represents the difference in distribution of lexical items in a specialized corpus and in a general one [1]. We need such general corpus, where the product features are weird. Weirdness is computed as follows:

$$weirdness(c) = \frac{freq_{special}\ (c)}{freq_{general}\ (c)},$$

where *special* means a specialized corpus and *general* is a general corpus. In our case, a specialized corpus is a collection of reviews.

TF-IDF

TF-IDF stands for the Term Frequency Inverse Document Frequency. It is a well-known feature that can be computed in a number of ways. In this work, we use the following formulae:

$$TFIDF(c,d) = TF(c,d)IDF(c),$$
$$TF(c,d) = freq_d(c),$$
$$IDF(c) = log\left(\frac{N(d)}{N(d_c)}\right),$$

where d is a document, d_c is a document with the candidate c.

It is important to note that TF-IDF depends nonlinearly on the size of the corpus, unlike the previously mentioned features.

PMI

The Pointwise Mutual Information (PMI) between two lexical items is a measure of the degree of statistical dependence between them and is defined as follows:

$$PMI(c,l) = log\left(\frac{freq(c,l)}{freq(c)freq(l)}\right),$$

where c is a candidate, l is some lexical item (word or phrase). $freq(c,l)$ is a frequency of them occurring together. It may mean, for example, occurrence one by one, in one sentence, in one document etc. We use different types of occurrences in this work.

3. Experiments

3.1. Dataset

We create and label a dataset with product features. We make an assumption that product features are single nouns and they explicitly appear in the text. This means that we consider only a part of the product feature name if it is a multi-word noun phrase. The side-effect is that representation of product features in a single noun may become ambiguous and hard to understand without context. However, the type of the product is known in advance and provides the context for disambiguation. We don't consider implicit product features [12] due their complex nature; however, they occur rarely because people usually use explicit descriptions to mention a product feature.

We extract all nouns from the reviews dataset described in [6]. It consists of 810 laptop reviews crawled from on-line shopping site Citilink[1]. The nouns were extracted and normalized using Mystem[2] part of speech tagger. It resulted in 1,994 unique nouns.

[1] http://www.citilink.ru/

[2] http://company.yandex.ru/technologies/mystem/

Then these nouns are manually labeled by 3 persons with 3 classes: a product feature (PF), not a product feature (NF) and a possible product feature (PPF). We agree to assume that a product feature is a product part, property or an attribute. All related entities and their parts are considered as well. For example, “keyboard”, “thickness” and “soft” are labeled as laptop features; “air”, “consumer”, “moment” are labeled as non-features; “resource”, “brain”, “glue” are labeled as possible product features.

The PPF class is hard to work with because it is very uncertain. It can be interpreted both as PF and NF. Depending on this, there will be different classification results and correlation agreement between the assessors. We will consider 3 solutions to this problem: remove all PPF, use them as PF and use them as NF.

The difficulty of product feature classification emerges already during the manual labeling process. Uncertainty and the lack of a formal feature definition result in low agreement between the assessors. The values of pair wise correlations between the assessors are 39%, 42% and 61% respectively. Considering PPF as PF produces even worse results: 32%, 32% and 45%. Considering PPF as NF gives correlations similar to the initial ones. If PPF is removed, then correlations are 61%, 62% and 97%. Such dispersion in agreements once again proves the difficulty of the work with product features.

Nine datasets for classification experiments are created from the mentioned labeled dataset. Three different approaches to treat the assessors’ agreement are used: intersection of labels, voting and the author’s labels. PPF label is assigned if there are 3 different votes. The three mentioned ways are applied to treat PPF label. Additional datasets are constructed for extra experiments.

Data imbalance is dealt both with oversampling the minority class and undersampling the majority class. Oversampling is performed in two rounds with a synthetic minority over-sampling technique (SMOTE) [13]. Each round doubles the minority class data. Then the order of instances is randomized. Undersampling is performed by means of removing instances of the majority class in order to make it the same size as the minority class.

3.2. Computation of classification features

We relied on the Yandex[3] search index as on the corpus for computing statistics, because it is supposed to be the biggest and all-embracing and, thus, the most precise from the freely available. The service YandexXML[4] provides a query API to the search engine. It has a limitation of the number of queries per day. The query result contains various fields, out of which we are interested in “found-docs”. It means an approximate number of documents relevant to the query. A simple software for making such queries has been written.

The mentioned approach has a number of restrictions. One cannot accurately argue, what is considered as a document, what percentage of document text is indexed,

[3] http://www.yandex.ru/

[4] http://xml.yandex.ru/

how the relevancy is computed, how precise the approximate number of documents is, etc. The search index is constantly changing and this puts certain restrictions on repeatability of our experiments. Another important issue is that it is impossible to compute pure statistics because the size of the index is unknown. As we mentioned earlier, there are some features that depend linearly (or on a constant) on the size of a corpus. In this case, we can deal with the unknown size of the corpus by means of normalization. However, TF-IDF depends non-linearly and we have to compute pseudo TD-IDF.

Let us consider the practical aspects of feature computation.

Frequency

We decide to use two different frequencies. The first is computed by means of Yandex Market[5] and represents a review corpus. The second is computed by means of Yandex and represents the whole Internet. We use the number of relevant documents returned for the queries "candidate host:market.yandex.ru" and "candidate". As mentioned earlier, there is no need to know the size of the Yandex Market and Internet corpora to compute frequencies.

Opinion word proximity

Yandex has quite a few query parameters that allow creating rather complex queries. One can search for the keywords occurrence in the same sentence and for the keywords occurring together not farther than a given number of words. We use two opinion words, "bad" and "good". Opinion word proximity is computed as the number of documents returned by the query "candidate /3 (good | bad)". This means that "good" or "bad" must be no farther than 3 words from the phrase "candidate". We will refer to it as to "OpinionNEAR3".

Weirdness

We employ two general purpose corpora: the Internet library lib.rus.ec[6] (LIB) and the Russian National Corpus[7] (RNC). LIB contains predominantly fiction and its size is 257,000 books. From RNC, the newspaper corpus is used that contains 332,720 documents (173,521,766 words). These corpora have been chosen because they are able to provide reasonable weirdness for the laptop product features. Weirdness-LIB is computed using the number of documents returned by the "candidate host:market.yandex.ru" and "candidate" queries. The software for querying RNC has been written. It returns the number of keyword occurrences and the number of documents with a keyword. Weirdness-RNC is computed using the mentioned numbers and the number of documents returned by the query "candidate host:market.yandex.ru". Frequencies from both general corpora are included as classification features as well. Interestingly, RNC provides a number of different sub-corpora and returns precise statistics. This is an area for further investigation.

---

[5] http://market.yandex.ru/

[6] http://lib.rus.ec/

[7] http://www.ruscorpora.ru/en/index.html

TF-IDF

The TF part of TF-IDF is computed as the number of documents returned by the query "candidate host:market.yandex.ru". The IDF part is computed using general corpora, as proposed in [4]. IDF-LIB cannot be computed precisely because the total number of documents is unknown. We use the number of books instead of it. IDF-RNC can be computed precisely because all the needed statistics is returned by RNC. The number of documents returned by RNC is used as a separate feature. We will refer to TF-IDF computed with LIB as to "TF-IDF-LIB" and to the one computed with RNC as to "TF-IDF-RNC".

PMI

We compute PMI with respect to the word "laptop" and a candidate. We try two different approaches to estimate . We use the number of documents returned by "candidate && laptop", that means search for both keywords in the same sentences. The second approach is to use the number of documents returned by "candidate & laptop", that means search for both keywords in the same documents. We perform search in Yandex and Yandex Market. Eventually, we have four versions of and 4 PMI consequently: "PMI-snt", "PMI-doc", "PMI-YM-snt", and "PMI-YM-doc".

We add the value 0.5 to the document count if it is used in logarithm or as a denominator. Finally, we have 23 different features including the assessors' labels.

3.3. Product feature classification

We use Weka[8] data mining tool [8] to conduct classification experiments. We chose 3 different classifiers: logistic regression, a decision tree and support vector machines (SVM). "J48" implementation of C 4.5 decision tree and "SMO" implementation of SVM is used. Logistic regression and the decision tree are run with default parameters. SVM is used with "data standardization", "build logistic models" and parameters. Two kernel types are set: radial basis (RBF) and normalized polynomial.

As we mentioned earlier, we prepared nine datasets. Table 2 reports classification results for 3 out of 9 prepared datasets and 1 additional one. These are the datasets created with the use of voting and with 3 different approaches to treat the possible product feature (PPF) class. "Vote-strong" dataset does not contain any converted PPF instances. All PPF labels were converted to non-product features (NF) in the "Vote-negative" and to product features (PF) in the "Vote-positive". "Vote-negativeO" is an oversampled "Vote-negative". The properties of these datasets are listed in Table 1. We conduct experiments with all remaining 6 datasets as well. They behave similarly to the "Vote-strong" classification and thus we didn't put them into the resulting table. The experiments were performed with 10-fold cross validation. The results in the table are the averages. Confidence interval for the F1-measure is similar for all experiments and is no more than 0.02 (alpha is 0.01). SVM column contains the best result of two kernels.

[8] http://www.cs.waikato.ac.nz/ml/weka/

Table 1. Properties of selected datasets

	PF	NF	total
Vote-strong	367	837	1204
Vote-negative	367	1627	1994
Vote-positive	1157	837	1994
Vote-negative^{O}	1468	1627	3095

Table 2. Aspect classification results

	Decision Tree			SVM			Logistic Regression		
Dataset	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Vote-strong	0.757	**0.711**	**0.733**	**0.801**	0.624	0.700	0.785	0.619	0.692
Vote-negative	0.509	**0.316**	0.390	**0.679**	0.294	**0.411**	0.609	0.259	0.363
Vote-positive	**0.790**	0.728	0.758	0.702	0.828	**0.760**	0.688	**0.831**	0.753
Vote-negative^{O}	**0.819**	**0.841**	**0.830**	0.816	0.766	0.790	0.785	0.727	0.755

Classification performance is quite good for the first dataset because it doesn't contain possible product features about which the assessors were not sure. The second dataset is imbalanced and the results are unsurprisingly mediocre. Interestingly, "Vote-positive" shows good performance despite the low agreement between the assessors. One of the reasons for this is that the real amount of single noun product features in our dataset may be comparable to the real amount of "neutral" or non-product feature nouns. The reason why the assessors did not agree on this was ambiguity of the nouns. Classification of the oversampled "Vote-negative" dataset provides the best results. We also conduct experiments with the undersampled "Vote-negative" and it performs very similarly to the first one, which is reasonable.

Different classifiers perform more or less as expected. SVM wins on the hardest imbalanced data, however due to some parameter tuning. The decision tree performs well on everything except the mentioned imbalanced data. In general, the classification results show applicability of the proposed approach to the product feature extraction. They also show that the possible product feature class can be considered both as a feature and as a non-feature. It may depend on the user's requirement: show more uncertain features or only precise ones.

Interestingly, our results are comparable to the results reported in papers on product features extraction for English [2], [9], [10], and [12]. They report an average F1-measure ranging from 0.76 to 0.86.

We are also interested to find out, which classification features are the most useful. We conduct experiments with each feature separately, but some of them produced zeros. We decide to combine at least two features instead. PMI is chosen as a default feature because it was used as a base feature in a similar work for English [2]. We have 2 modifications of PMI: "PMI-snt" and "PMI-doc". Experiments with a pairwise combination of different features with them are performed. SVM classifier is used with the same settings as mentioned previously and RBF kernel. The results are represented in Table 3.

Table 3. Aspect classification results with different features

	PMI-snt			PMI-doc		
Feature	**P**	**R**	**F1**	**P**	**R**	**F1**
TF-IDF-LIB	0.643	0.172	0.271	0.610	**0.226**	**0.330**
Weirdness-LIB	0.778	0.038	0.073	**0.789**	0.041	0.078
Weirdness-RNC	0.321	0.025	0.046	0.529	0.025	0.047
TF-IDF-RNC	0.344	0.030	0.055	0.481	0.071	0.124
PMI-YM-docs	0.383	0.049	0.087	0.154	0.005	0.011
PMI-YM	0.242	0.022	0.040	0.278	0.014	0.026
OpinionNEAR3	0.231	0.016	0.031	0.512	0.060	0.107

One can see that the “TF-IDF-LIB” and “Weirdness-LIB” are the most useful features in combination with PMI. Interestingly, TF-IDF and Weirdness computed with a different general corpus provide worse results. It is accounted for by the use of the newspaper corpus from RNC, while the corpus in LIB is mostly fiction. Newspapers are more probable to have product features, rather than fiction. Another interesting observation is that PMI computed using “the same document” (“PMI-doc”) query perform slightly better than the one computed with “the same sentence” query (“PMI-snt”).

4. Conclusion

We performed the task of product features extraction from Russian reviews. It was addressed as a classification problem. A product feature dataset was created and labeled. A number of different classification features were used and several classification algorithms applied. The experiments demonstrated efficiency of our approach.

Our further work is to use additional linguistic and statistical attributes for classification. Spelling corrector will be employed to correct the spelling of candidates. We plan to apply sequence labeling classifiers as well. We will do product features clustering to group them into meaningful groups. This may help us to filter features as well.

References

1. *Ahmad, K.; Gillam, L.; and Tostevin, L.* 1999. University of surrey participation in trec8: Weirdness indexing for logical document extrapolation and retrieval (wilder). In The Eighth Text Retrieval Conference (TREC-8).
2. *Ana M. Popescu, Oren Etzioni.* Extracting Product Features and Opinions from Reviews. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (2005)
3. *Brody, S. and S. Elhadad.* An Unsupervised Aspect-Sentiment Model for Online Reviews. In Proceedings of The 2010 Annual Conference of the North American Chapter of the ACL, 2010.

4. *Chatviorkin Ilya, Lukashevich Natalia*. Automatic Extraction of Domain-Specific Opinion Words. Proceedings of the International Conference Dialog, 2010.
5. *Chetviorkin I. I., Braslavski P. I.* Sentiment analysis track at ROMIP 2011. Dialog 2011.
6. *Chugreev A., Marchuk A., Makeev I., Mokaev T., Skudarnov Y., Bat'kovich D., Ulanov A.* GoodsReview — A Service for Automatic Sentiment Analysis [GoodsReview — servis avtomaticheskogo analiza portrebitel'skogo mneniya]. Proceedings of KESW, 2012.
7. *Ghani, R., K. Probst, Y. Liu, M. Krema, and A. Fano.* Text mining for product attribute extraction. ACM SIGKDD Explorations Newsletter, 2006, 8(1): p. 41-48.
8. *Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I. H.* (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.
9. *Hu, M. and B. Liu*. Mining and summarizing customer reviews. In Proceedings of ACM SIGKD International Conference on Knowledge Discovery and Data Mining (KDD-2004), 2004.
10. *Jakob, N., & Gurevych, I.* (2010, October). Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (pp. 1035–1045). Association for Computational Linguistics.
11. *Jin, W. and H. Ho.* A novel lexicalized HMM-based learning framework for web opinion mining. In Proceedings of International Conference on Machine Learning (ICML-2009), 2009.
12. *Liu, Bing.* Web Data Mining. Exploring Hyperlinks, Contents, and Usage Data. 2nd ed. 2011, XX, 622 p.
13. *Nitesh V. Chawla* et. al. (2002). Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research. 16:321–357.
14. *Pak A., Paroubek P.* Language independent approach to sentiment analysis (LIMSI Participation in ROMIP'11), Dialog 2011.
15. *Riloff E*. Automatically constructing a dictionary for information extraction tasks. Proceedings of the National Conference on Artificial Intelligence, 811–811, 1993
16. *Titov, I. and R. McDonald.* Modeling online reviews with multi-grain topic models. In Proceedings of International Conference on World Wide Web (WWW-2008), 2008.
17. *Wu, Y., Q. Zhang, X. Huang, and L. Wu*. Phrase dependency parsing for opinion mining. In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2009), 2009.
18. *Zhang Z., Iria J., Brewster C., Ciravegna F. A.* Comparative Evaluation of Term Recognition Algorithms. In the sixth international conference on Language Resources and Evaluation, (LREC 2008).