МЕТОДЫ УЛУЧШЕНИЯ ЧИТАБЕЛЬНОСТИ ПРИ АВТОМАТИЧЕСКОЙ КОНТЕКСТУАЛИЗАЦИИ НОВОСТНОГО ТВИТА

IMPROVING READABILITY OF AUTOMATIC NEWS TWEET CONTEXTUALIZATION

Ермакова Л.М. (liana.ermakova.87@gmail.com), Пермский государственный национальный исследовательский университет, Пермь, Россия

Мот Ж. (josiane.mothe@irit.fr), Institut de Recherche en Informatique de Toulouse, Тулуза, Франция

Овчинникова И.Г. (ira.ovchi@gmail.com), University of Haifa, Израиль - Пермский государственный национальный исследовательский университет, Пермь, Россия

Ключевые слова: автоматическое сводное реферирование, квазиреферирование, сглаживание по локальному контексту, задача о рюкзаке, задача коммивояжёра

Ermakova L.M. (liana.ermakova.87@gmail.com), Perm State University, Perm, Russia

Mothe J. (josiane.mothe@irit.fr), Institut de Recherche en Informatique de Toulouse, Toulouse, France

Ovchinnikova I.G. (ira.ovchi@gmail.com), University of Haifa, Israel – Perm State University, Russia

The main objective of this research is to provide context for news tweets based on English Wikipedia. To this end a multi-document summarization system was implemented. Sentence extraction was performed on the basis of TF-IDF measure enriched by smoothing from local context, named entity recognition and part-of-speech weighting. The hypothesis was that the remoteness decreases the influence of the context on the target sentence sense. We improved this baseline approach by anaphora resolution which enhanced not only readability, but also relevance. Extraction was modeled as a rucksack problem where the number of words corresponds to weight and value is represented by F-measure of readability and relevance. Sentences were modeled as graph vertex and the similarity measure between them corresponded to edges. So sentence reordering was reduced to travelling salesman problem which was solved by greedy algorithm. To avoid redundancy we used threshold of the similarity of noun set of sentences.

Key words: automatic multi-document summarization, extracts, smoothing from local context, travelling salesman problem, rucksack problem

1. Introduction

Many people follow news in Twitters. However, tweets are short and they may include information which is not understandable to user without some context (e.g. user may be not familiar with mentioned named entities like persons, organizations or places). The idea to contextualize tweets is quite recent [1]. Moreover, in [1] the system provides to a user only references to Wikipedia pages, not a concise summary. Therefore the main objective of this research is to provide a context for news tweets based on English Wikipedia. To this end an automatic extraction system is implemented. The system provides a summary of predefined number of words for a given tweet. 132 tweets were considered. A tweet included the title and the first sentence of a New York Times articles.

Summary is a “condensed version of a source document having a recognizable genre and a very specific purpose: to give the reader an exact and concise idea of the contents of the source” [2]. Summaries are either “extracts”, if they contain the most important sentences extracted from the original text, or “abstracts”, if these sentences are paraphrased [3][4][5]. The majority of existing abstract generation systems have extraction component [4]. In general, extracts have low readability [6]. However, if a text extraction system deals with entire sentences, locally they may have higher readability than generated phrases since they are written by humans. Therefore, it is important to increase the global readability of extracted passages.

As a baseline system we used an extraction component developed for INEX 2011 [7]. Extraction is performed on the basis of TF-IDF measure enriched by smoothing from local context, named entity recognition and part-of-speech (POS) weighting. Traditionally, smoothing from local context does not take into account the remoteness of the target sentence [8]. Per contra, we believe, that as the distance increases, the influence of the context on the target sentence goes down. The system was compared to other 11 systems and it showed the best results in relevance evaluation [9]. However, there are several drawbacks in readability: unresolved anaphora and sentence ordering. One of the ways to resolve anaphora is to extract previous sentence. We decided not to do it since it decreases relevance. Instead, we resolved pronoun anaphora by adding the mention from the context. In single document summarization sentence ordering is not so crucial as far as it may be done by using initial relative order in the original text. However, it is impossible for multi-document summarization. We propose an approach to increase global coherence of text on the basis of its graph representation. The hypothesis is that neighboring sentences should be somehow similar. Computing the similarity between sentences allowed reducing sentence ordering task to travelling salesman problem which may be solved by greedy nearest neighbor algorithm. To avoid redundancy we mapped sentences into a noun set and rejected duplicate phrases by a predefined threshold (by defaults 70%). It may happen that a sentence has no sense in the given context. In this case it is better to replace it by another sentence. Therefore, the system deals with sentences having twice more words than it is set by a user. Let be the number of words set by a user. After selecting the most relevant sentence of words in total, we applied the branch and bound method to this rucksack problem. As weight we considered the number of words in a sentence, and the F-measure of relevance and readability represented value.

2. Overview

The processing information in tweets is based on the definition of a topic. As usual, the goal is searching in the Internet for texts with the same topic and summarizing the result of the search. Actually, we have two problems to solve: (1) the basis for query and (2) the way to summarize and represent results. From a linguist's point of view the problems deal with linguistic units to search and discourse (or rhetoric) features of the texts to process and to generate as a summary.

Several systems can automatically discover the wide-range of vocabulary used in tweets, including topic tags, and they use linguistic processing to collect and summarize the thousands of ways people have of saying the same thing (ex. gr., Linguamatics). However, necessity to recognize named entities, anaphora and usage of slang create misunderstanding and difficulties in topic recognition [10]. Also labeling the POS in a tweet enables complex analysis [11]. Named entities recognition and POS labeling are essential for genre identification of texts which are probably linked to tweets. Genre correlates with topic, although the correlation is complicated and enables complex analysis of words and syntactic constructions. Nevertheless, for summary the genre of the resource could be the factor of great importance. For example, while searching for texts with named entities connected to current political events, references to historical or education texts are irrelevant. The relevant processing should be lexical / contextual based and sensible to discourse (genre) peculiarities at the same time. As a restriction to prevent irrelevant results we can apply grammar filters such as POS distribution and syntactic constructions [12]. Looking for characterization of named entities, we can apply a noun phrase (NP) with named entity as N + the verb *to be* + Adj. Several researchers tried to determine various semantic properties of verbs automatically [13][14]. These approaches, however, attempt to determine properties of verbs viewed in isolation and do not deal with sequences of verbs and verboids in the context. In the case of searching for events related to the named entity, a sequence of verbs and verbal forms in the left context to the named entity will play a role of a suitable filter [15]. In the system for Automatical Genre Classification (AGC) the subset of POS are used in order to maintain performance across changes in the topical distribution [16]. Thanks to subset of POS we can eliminate reference to irrelevant genres. Thus, in our research we combine lexical / semantic, grammar and genre properties of text in searching relevant connections with tweet.

Let's discuss the summary generation. Automatically summarizing based on lexical / contextual approach are implemented in SUMMARIST system, which generates key-words and extracts. Meanwhile the problem of summary generation is still unsolved. Genre is the key peculiarity of the text for summarization. The better documents' structure, the more transparent the content of document's parts and their connections, the better the system for their automatic summarization. That is why such texts as news articles, medical documents, legal documents, and papers in the area of computer science are the most popular material for summarization [17]. By the way, the material is relevant for tweets processing.

3. Method description

3.1.Preprocessing

Firstly we looked for the documents similar to the queries. The hypothesis was that relevant sentences come from the relevant documents [8]. For this stage, document retrieval was performed by the Terrier Information Retrieval Platform (http://terrier.org/), an open-source search engine developed by the School of Computing Science, University of Glasgow. To this end we transformed queries into to the format accepted by Terrier.

The next stage was parsing of tweets and retrieved texts by Stanford CoreNLP developed by the Stanford Natural Language Processing Group. CoreNLP integrates such tools as POS tagger, named entity recognizer (NER), parser and the co-reference resolution system (http://nlp.stanford.edu/software/corenlp.shtml). It produces an XML document as an output. It uses the Penn Treebank tag set [18]. In our approach, tweets were transformed into queries with POS tagging and recognized named entities (NE). It allows taking into account different weights for different tokens within a query, e.g. NE are considered to be more important than common nouns; nouns are more significant than verbs; punctuation marks are not valuable, etc.

3.2.Sentence Retrieval

After preprocessing step we computed indices for each section and each sentence. The index included not only word forms and lemmas, but also NE.

The general idea of the proposed approach is to compute similarity between the query and sentences and to retrieve the most similar passages. To this end we used the standard TF-IDF measure. We extended this approach by adding weight coefficients to POS, NE, headers, sentences from abstracts, and definitional sentences. Moreover, sentence meaning depends on the context. Therefore we used an algorithm for smoothing from the local context which will be described later. The sentences were sorted by their similarity scores. The sentences with the highest score were added to the summary until the total number of words exceeds the predefined value (by default 500).

In the implemented system there is a possibility to choose one of the following similarity measures [19]:

1. Cosine similarity:

(1)

2. Dice similarity:

(2)

3. Jaccard similarity:

(3)

where is a query, is a sentence, is the occurrence of the i-th token in a query and is the occurrence of the i-th token in a sentence. If the token is not presented in the query or in the sentence, or is equal to 0 respectively.

We took into account only lexical vocabulary overlap between a query and a sentence. However it is possible also to consider morphological and spelling variants, synonyms, hyperonyms, etc.

3.2.1. Weighting

Word weighting is applied to improve performance, e.g. usually it is better not to take into account stop-words. Our system provides several ways to assign score to words. The first option is to identify stop-words by frequency threshold. The second way is to assign different weights to different parts of speech.

Besides, NE comparison is hypothesized to be very efficient for contextualizing tweets about news. Therefore for each NE in queries we searched corresponding NE in the sentences. If it is found, the whole similarity measure is multiplied by NE coefficient computed by the formula:

(4)

where is floating point parameter given by a user (by default it is equal to 1.0), is the number of NE appearing in both query and sentence, is the number of NE appearing in the query. We used Laplace smoothing to NE by adding one to the numerator and the denominator. The sentence may not contain a NE from the query and it can be still relevant. However if smoothing is not performed the coefficient will be zero. NE recognition is performed by Stanford CoreNLP. We considered only the exact matches of NE. Synonyms were not identified. However, it may be done later applying WordNet, which includes major NE.

We assigned lower weights to sentences without personal verbs since this kind of sentences (e.g. labels, headers etc.) decreases text fluency although they seem to be very informative.

We assumed that definitional sentences are extremely important to contextualizing task. Therefore they should have higher weights. We have taken into account only definitions of NE by applying the following linguistic pattern:

is a personal form of the verb *to be*. Noun phrase recognition is also performed by Stanford parser. We considered only sentences that occurred in abstracts since they contain more general and condensed information and usually include definitions in the first sentence. However, the number of extracted definitions was quite small and therefore we did not use them in our runs.

3.2.2. Smoothing from Local Context

Traditionally, sentences are smoothed by the entire collection, but there exist another approach namely smoothing from local context. We believe that the major drawback of this approach is that the same weight is assigned to all sentences from the context [8]. In contrast, we assume that the importance of the context reduces as the distance increases. So, the nearest sentences should produce more effect on the target sentence sense than others. For sentences with the distance greater than k this coefficient is zero. The total of all weights should be equal to one.

The system allows taking into account k neighboring sentences with the weights depending on their remoteness from the target sentence. In this case the total target sentence score is a weighted sum of scores of neighboring sentences and the target sentence itself:

(5)

(6)

(7)

where is a target sentence weight set by a user, are weights of the sentences from k context. The weights become smaller as the remoteness increases. If the sentence number in left or right context is less than k, their weights are added to the target sentence weight . This allows keeping the sum equal to one. By default, , and the target sentence weight is equal to 0.8.

3.3.Readability improvement

Previous steps were performed to improve relevance of extracted sentences, but the other important aspect of extracts is readability. The baseline system had two major drawbacks in readability: unresolved anaphora and sentence ordering.

One of the ways to resolve anaphora is to include the previous sentence in extract. We decided not to do it since it decreases relevance. Instead, we resolved pronoun anaphora by adding the mention from the context. This resolution was also taken into account during relevance computing. If the representative is in the same sentence as a pronoun it should not be added. Anaphora resolution was performed by Stanford CoreNLP.

In single document summarization sentence ordering is not so crucial as far as it may be done by using initial relative order in the original text. However, it is impossible for multi-document summarization. We propose an approach to increase global coherence of text on the basis of its graph representation. The hypothesis is that neighboring sentences should be somehow similar to each other and the total distance between them should be minimal. So, we computed the similarity between sentences and reduced sentence ordering task to travelling salesman problem. To solve it we applied greedy nearest neighbor algorithm with minor changes, namely we tried every vertex as the start one and chose the best result. If two relevant sentences are neighbors in the original text, they should be considered as a single vertex. The major disadvantage of this approach is that a text with the same repeated sentence would be falsely overscored. The naïve approach to avoid it is to use a threshold value. However, it cannot deal with sentences of almost the same sense but different length (e.g. with more adjectives). As far as nouns provide much semantics, a sentence may be mapped into a noun set. We applied this mapping and the threshold. It may happen that a sentence has no sense in the given context. In this case it is better to replace it by another sentence. Therefore, the system deals with sentences having twice more words than it is set by a user . After selecting the most relevant sentence of words in total, we applied the branch and bound method to this rucksack problem. As weight we considered the number of words in a sentence, and the F-measure of relevance and readability represented value.

4. Evaluation

Evaluation was performed by the FRESA package [9]. Summaries were compared with the original NYT articles (see Table 1) and with the pool of relevant passages obtained manually (see Table 2). The simple log difference (8) was used, since the Kullback-Leibler divergence was too sensitive to smoothing on the given collection [9]:

(8)

The system was compared with other 11 systems and it obtained the best results for relevance. 7 of these systems were based on Indri [8].

Table 1. Log difference between the summaries and original NYT articles

№	RUN	RANKING	UNIGRAM	BIGRAM	WITH 2-GAP	AVERAGE
1.	***ID12RIRIT_05_2_07_1_jac***	***0.104925***	***0.0447***	***0.076644***	***0.104925***	***0.076629***
2.	***ID12RIRIT_07_2_07_1_dice***	***0.104933***	***0.044728***	***0.076659***	***0.104933***	***0.076646***
3.	***ID12RIRIT_default***	***0.104937***	***0.044739***	***0.076668***	***0.104937***	***0.076653***
4.	ID129RRun1	0.10604	0.045626	0.077687	0.10604	0.077664
5.	ID132RRun1	0.106118	0.046187	0.077947	0.106118	0.077946
6.	Baselinesum	0.10646	0.046049	0.078101	0.10646	0.078084

7.	ID126RRun1	0.106536	0.045998	0.078113	0.106536	0.078101
8.	ID128RRun2	0.106601	0.046065	0.078179	0.106601	0.078167
9.	ID138RRun1	0.106605	0.046122	0.078225	0.106605	0.078201
10.	ID129RRun2	0.10708	0.046751	0.078775	0.10708	0.078746
11.	ID129RRun3	0.107209	0.046798	0.078864	0.107209	0.078837
12.	ID126RRun2	0.10728	0.046852	0.078916	0.10728	0.078897
13.	ID128RRun3	0.107341	0.046872	0.07895	0.107341	0.078937
14.	ID123RI10UniXRun1	0.107491	0.047084	0.079149	0.107491	0.079121
15.	Baselinemwt	0.10766	0.047508	0.079385	0.10766	0.079387
16.	ID62RRun1	0.10769	0.047283	0.079344	0.10769	0.079319
17.	ID128RRun1	0.107911	0.047482	0.07955	0.107911	0.079529
18.	ID62RRun3	0.107969	0.047598	0.079638	0.107969	0.079614
19.	ID62RRun2	0.107993	0.047674	0.079689	0.107993	0.079662
20.	ID123RI10UniXRun2	0.108036	0.047735	0.07973	0.108036	0.07971
21.	ID123RI10UniXRun3	0.108681	0.048326	0.080369	0.108681	0.080337
22.	ID46RJU_CSE_run1	0.108948	0.0487	0.080679	0.108948	0.08065
23.	ID46RJU_CSE_run2	0.10895	0.048702	0.08068	0.10895	0.080651
24.	ID124RUNAMiiR12	0.109389	0.04931	0.081181	0.109389	0.081161
25.	ID124RUNAMiiR3	0.110429	0.050541	0.082313	0.110429	0.082288

Table 2. Log difference between summaries and the pool of relevant sentences

№	Run	Ranking	unigram	bigram	with 2-gap	Average
1.	***ID12RIRIT_default***	***0.105506***	***0.048639***	***0.07867***	***0.105506***	***0.078697***
2.	***ID12RIRIT_07_2_07_1_dice***	***0.105747***	***0.048781***	***0.078857***	***0.105747***	***0.07889***
3.	***ID12RIRIT_05_2_07_1_jac***	***0.106195***	***0.049083***	***0.079249***	***0.106195***	***0.079277***
4.	ID129RRun1	0.107806	0.050253	0.080676	0.107806	0.080689
5.	ID129RRun2	0.110616	0.05178	0.082987	0.110616	0.082954
6.	ID128RRun2	0.111033	0.052372	0.08345	0.111033	0.083438
7.	ID138RRun1	0.111516	0.052383	0.08374	0.111516	0.083716
8.	ID132RRun1	0.111666	0.052567	0.083836	0.111666	0.083857
9.	ID126RRun1	0.112529	0.053464	0.084754	0.112529	0.084752
10.	Baselinesum	0.114346	0.053691	0.085915	0.114346	0.085881
11.	ID126RRun2	0.114404	0.054608	0.086328	0.114404	0.086311
12.	ID128RRun3	0.11512	0.054904	0.086875	0.11512	0.086846
13.	ID129RRun3	0.115219	0.054883	0.086928	0.115219	0.086896
14.	ID46RJU_CSE_run1	0.115557	0.056092	0.087656	0.115557	0.087617
15.	ID46RJU_CSE_run2	0.11558	0.056122	0.087682	0.11558	0.087643
16.	ID62RRun3	0.117158	0.056456	0.088684	0.117158	0.088667
17.	ID123RI10UniXRun2	0.117196	0.056143	0.088538	0.117196	0.088537
18.	ID128RRun1	0.117406	0.05655	0.088886	0.117406	0.088852
19.	Baselinemwt	0.117854	0.055786	0.088604	0.117854	0.088701
20.	ID62RRun1	0.118016	0.05661	0.089207	0.118016	0.089203
21.	ID123RI10UniXRun1	0.118346	0.056717	0.08948	0.118346	0.08945
22.	ID62RRun2	0.118805	0.057196	0.089971	0.118805	0.089925
23.	ID124RUNAMiiR12	0.122111	0.060737	0.09335	0.122111	0.093325
24.	ID123RI10UniXRun3	0.123938	0.061052	0.094556	0.123938	0.094502

25.	ID124RUNAMiiR3	0.124792	0.062794	0.095747	0.124792	0.095726

However, the readability should be improved (see Table 3).

Table 3. Readability score

№	Run	Score	№	Run	Score
1.	ID129R_Run1	359.0769	13.	ID126R_Run2	296.3922
2.	ID129R_Run2	351.8113	14.	ID62R_Run2	288.6154
3.	ID126R_Run1	350.6981	15.	ID128R_Run1	284.4286
4.	ID46R_JU_CSE_run1	347.92	16.	ID62R_Run3	277.9792
5.	***ID12R_IRIT_05_2_07_1_jac***	***344.1154***	17.	ID62R_Run1	266.1633
6.	***ID12R_IRIT_default***	***339.9231***	18.	ID18R_Run1	260.1837
7.	***ID12R_IRIT_07_2_07_1_dice***	***338.7547***	19.	ID123R_I10UniXRun1	246.9787
8.	ID128R_Run2	330.283	20.	ID123R_I10UniXRun2	246.5745
9.	ID46R_JU_CSE_run2	330.14	21.	ID123R_I10UniXRun3	232.6744
10.	ID129R_Run3	325.0943	22.	ID124R_UNAMiiR12	219.1875
11.	ID138R_Run1	306.2549	23.	Baseline_mwt	148.2222
12.	ID128R_Run3	297.4167	24.	ID124R_UNAMiiR3	128.3261

The readability evaluation was performed manually. Assessors should indicate if a passage contained one of the following drawbacks:

- The passage has syntactical problems (e.g. bad segmentation).
- The passage contains an unresolved anaphora.
- The passage has redundant information (that is to say, the information is already mentioned).
- The passage is meaningless in the given context.

The score of a summary was the average normalized number of words in valid passages [9].

According to evaluation results the most significant drawback was unresolved anaphora. Therefore it was the first thing to fix. We evaluated new summaries and human judgment provides evidence that there are fewer problems with anaphors and sentence ordering.

5. Conclusion

In this paper we describe a method of tweet contextualization on the basis of the local Wikipedia dump.

Firstly, we looked for relevant Wikipedia pages using the search engine Terrier. Secondly, the input tweets and the found documents were parsed by Stanford CoreNLP. After that, a new index for sentences was constructed. It includes not only stems but also NE. Then we searched for relevant sentences. To this end similarity between the query and sentences was computed using an extended TF-IDF measure. Weight coefficients to POS, NE, headers, sentences from abstracts, and definitional sentences were added. Moreover, the algorithm for smoothing from local context is provided. We assume that the importance of the context depends on the remoteness from the target sentence. So, the nearest sentences should produce more effect on the target sentence sense than others. Remote sentences (with the distance greater than k) should not be taken into account.

We enhanced the baseline system by anaphora resolution. It produced effect not only on readability, but also improved the relevance.

Readability was improved by modeling sentence extraction as a rucksack problem where the number of words corresponds to weight and value is represented by F-measure of readability and relevance. A travelling salesman problem was applied to sentence reordering. Sentences were modeled as graph vertex and the similarity measure between them corresponded to edges. To avoid redundancy we used threshold to the similarity of noun set of sentences.

The ongoing work is on redundancy treatment. The idea is to make a mapping not in the set of nouns, but in the set of synsets. Synonyms may be also used to relevance computing. Another way of system improvement is to analyze the subject-predicate relation.

References

[1] H. Saggion and G. Lapalme, "Generating Indicative-Informative Summaries with SumUM," *Association for Computational Linguistics*, 2002.

[2] J. Vivaldi, I. da Cunha, and J. Ramırez, "The REG summarization system at QA@INEX track 2010," 2010.

[3] G. Erkan and D. R. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization," *Journal Of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.

[4] S. Gholamrezazadeh, M. A. Salehi, and B. Gholamzadeh, "A Comprehensive Survey on Text Summarization Systems," *Computer Science and its Applications*, pp. 1–6, 2009.

[5] D. R. Radev and K. R. McKeown, "Generating natural language summaries from multiple on-line sources," *Computational Linguistics - Special issue on natural language generation*, vol. 24, pp. 469–500, 1998.

[6] L. Ermakova and J. Mothe, "IRIT at INEX: Question Answering Task," *INEX 2011. Workshop Preproceedings*, pp. 160–166, 2011.

[7] V. G. Murdock, "Aspects of Sentence Retrieval," *Dissertation*, 2006.

[8] E. SanJuan, V. Moriceau, X. Tannier, P. Bellot, and J. Mothe, "Overview of the INEX 2011 Question Answering Track (QA@INEX)," *Focused Retrieval of Content and Structure, 10th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2011), Geva, S., Kamps, J., Schenkel, R. (Eds.)*, 2012.

[9] B. Han and T. Baldwin, "Lexical normalisation of short text messages: makn sens a #twitter," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, Portland, Oregon, 2011, pp. 368–378.

[10] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "Part-of-speech tagging for Twitter: annotation, features, and experiments," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, Portland, Oregon, 2011, pp. 42–47.

[11] L. Ermakova, "Spam and Phishing Detection in Various Languages," *International Journal "Information Technologies and Knowledge,"* vol. 4, no. 3, pp. 216–232, 2010.

[12] E. V. Siegel, "Corpus-based linguistic indicators for aspectual classification," in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, College Park, Maryland, 1999, pp. 112–119.

[13] P. Merlo, S. Stevenson, V. Tsang, and G. Allaria, “A multilingual paradigm for automatic verb classification,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, Pennsylvania, 2002, pp. 207–214.

[14] И. Г. Овчинникова, Л. Л. Черепанова, and Е. В. Ягунова, “Вариативность новостных текстов в аспекте информационного анализа,” *Проблемы динамической лингвистики: матер. Международной научн.й конф., посвященной 80-летию профессора Л.Н. Мурзина*, pp. 401–406, 2010.

[15] P. Petrenz and B. Webber, “Stable classification of text genres,” *Comput. Linguist.*, vol. 37, no. 2, pp. 385–393.

[16] A. Kazantseva and S. Szpakowicz, “Summarizing short stories,” *Comput. Linguist.*, vol. 36, no. 1, pp. 71–109, 2010.

[17] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, “Building a large annotated corpus of English: the Penn Treebank,” *Computational Linguistics*, vol. 19, 1993.

[18] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

.

.