Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2017"

Moscow, May 31—June 3, 2017

WHICH IR MODEL HAS A BETTER SENSE OF HUMOR? SEARCH OVER A LARGE COLLECTION OF JOKES

Bolotova V. V. (lurunchik@gmail.com),
Blinov V. A. (vladislav.blinov@urfu.ru),
Mishchenko K. I. (ki.mishchenko@gmail.com),
Braslavski P. I. (pbras@yandex.ru)

Ural Federal University, Yekaterinburg, Russia

This paper describes experiments on humorous response generation for short text conversations. Firstly, we compiled a collection of 63,000 jokes from online social networks (VK and Twitter). Secondly, we implemented several context-aware joke retrieval models: BM25 as a baseline, query term reweighting, word2vec-based model, and learning-to-rank approach with multiple features. Finally, we evaluated these models in two ways: on the community question answering platform Otvety@Mail.ru and in laboratory settings. Evaluation shows that an information retrieval approach to humorous response generation yields satisfactory performance.

Key words: computational humor, dialog systems, information retrieval approach, natural language processing

У КАКОЙ МОДЕЛИ ИНФОРМАЦИОННОГО ПОИСКА ЛУЧШЕ ЧУВСТВО ЮМОРА? ПОИСК В БОЛЬШОЙ КОЛЛЕКЦИИ ШУТОК

Болотова В. В. (lurunchik@gmail.com),
Блинов В. А. (vladislav.blinov@urfu.ru),
Мищенко К. И. (ki.mishchenko@gmail.com),
Браславский П. И. (pbras@yandex.ru)

Уральский федеральный университет,
Екатеринбург, Россия

1. Introduction

Following recent trends in the widespread use of dialog systems like Apple Siri, Microsoft Cortana, Google Now and others, it becomes important to incorporate sense of humor into them. Humorous responses can help to deal with out-of-domain queries which have become an issue for the chatbots. Moreover, jokes that occasionally appear during interaction can make appear dialog systems more human-like.

Sense of humor plays a significant role in human-computer interaction. In particular, (Nijholt, 2002; Khooshabeh et al., 2011) have shown that adding humor capabilities to conversational agents results in more trustable and attractive interaction for users. Furthermore, Nijholt (2002) has summarized research according to which a sense of humor is generally considered a valued characteristic of others and plays a significant role in some task-oriented interactions, e.g. teaching.

The aim of our study is to examine the effectiveness of information retrieval approach to humorous response generation. Firstly, we compiled a collection of 63,000 jokes from online social networks (VK and Twitter). Secondly, we implemented several context-aware joke retrieval models: BM25 as a baseline, query-term reweighting, word2vec-based, IBM model 1, and learning-to-rank approach with multiple features. Finally, we evaluated these model in two ways: on the community question answering platform Otvety@Mail.ru and in laboratory settings.

2. Related Work

There are two main research directions in computational humor: humor recognition and humor generation. Stock and Strapparava (2003) have considered the problem of generating funny expansions for known and unknown acronyms. For known acronyms the implemented system keeps some words unchanged (usually nouns) and finds contrasting but similarly sounding words for the remaining ones using WordNet and other linguistic resources. For unknown acronyms the system starts with a WordNet synset and generates a syntactically consistent but semantically incongruous sequence of words. Ritchie (2005) has systematized different types of puns and proposed mechanisms for automatic pun generation. Valitutti et al. (2013) have proposed a method how to make 'adult' puns from short text messages by lexical replacement. A related study (Hong and Ong, 2009) addresses the task of automatic template extraction for pun generation. The extracted templates consist of a syntax structure and binary relations between words (such as *SynonymOf*, *Compoundword*, *SoundsLike*, etc.). After the learning stage the authors obtained 27 templates. Best automatically generated jokes received about the same evaluation scores as the human ones.

The study (Mihalcea and Strapparava, 2006) proposes a method for adding a joke to an email message or a lecture note and is close to our approach. The solution exploits an automatically gathered collection of 16,000 one-liners. For a given text fragment the application finds the semantically closest joke using the latent semantic analysis (LSA). A small-scale users study showed good performance and reception of the proposed solution, though even returning a random joke provided relatively good performance (as an opposite to not adding any joke at all).

Yang et al. (2015) have drawn attention to *humor anchors*, i.e. words prompting comic effect, and have addressed the task of *humor anchor recognition*.

In the field of information retrieval, Friedland and Allan (2008) proposed a domain-specific joke retrieval model based on jokes structure and interchangeable word classes. Surdeanu et al. (2011) investigated usefulness of different linguistic features for search in large archives of questions and answers for non-factoid questions. The study does not deal with humorous content, but the approach is still similar to ours.

Ritter et al. (2011) studied the applicability of a data-driven approach for generating responses to Twitter status posts. They used phrase-based statistical machine translation while trying to solve the problem.

In our initial experiments (Blinov, 2016) we evaluated popularity-based ranking (Likes model). This model can be regarded as an analogue of query-independent ranking based on document authority (e.g. PageRank)—a funny joke is potentially still funny, even if it is not quite in the context. The model requires only minimal overlap between a question and candidate responses (one common noun or verb) and ranks the responses by descending normalized Like scores. However, evaluation showed that BM25 scoring outperforms simple joke popularity.

3. Data

3.1. Joke Collection

We gathered a collection of jokes from popular humor-related user communities and accounts on VK[1], the largest Russian online social network, and Twitter[2]. We collected posts without media content (images and video) that gained more than 500 "likes" for VK and at least 1 for Twitter (where "likes" are much rarer). The VK posts longer than 250 characters were eradicated. Table 1 summarizes the sources of the initial corpus.

Table 1. Initial collection of jokes by source

Community/Account	URL	Size
F*** Normality	https://vk.com/trahninormalnost1	70,647
Evil Incorporated	https://vk.com/evil_incorparate	69,431
Witty	https://vk.com/ostroym	42,267
Strange Humor	https://vk.com/c.umor	44,287
Humor FM	https://twitter.com/_humor_fm_	3,578
About Humor	https://twitter.com/abouthumor	332
Drunken Twitter	https://twitter.com/drunktwi	15,335
Caucasian Humor	https://twitter.com/kavhum	4,988
Funny Radio	https://twitter.com/veseloeradio	12,312
Jokes and Anecdotes	https://twitter.com/anecdot_eshe	5,181
	Total	268,358

[1] https://vk.com/

[2] https://twitter.com/

Out of those posts we retained only one-liners and two-turn dialog jokes (see *Examples 1* and *2*, respectively), 226,431 jokes total. Then, we removed duplicates based on similarity of lemmatized bag-of-words representations. This step reduced collection size drastically—down to 63,293 jokes.

Example 1:

Лекарства так подорожали, что скоро мы будем дарить их друг другу на день рождения... Чтобы дожить до следующего.....))))))

Drugs have become so expensive that we will soon present them like a birthday gift... To attain the next anniversary.....))))))

Example 2:

— Ты спать собираешься?
— Да, сейчас, закончу делать ничего и пойду.

— Are you going to go to bed?
— Yes, now, I finish doing nothing and go.

3.2. CQA Dataset

We also collected a large historical dataset of question-answer pairs from the Humor category of a popular Russian community question answering platform Otvety@Mail.Ru[3]. Each question there can be answered once by any user and each answer can be rated once by any user (Fig. 1 shows the user interface of the CQA platform). In addition, the asker can mark an answer as "the best" and then the question will be tagged as "solved". The collection that we compiled consists of more than 35,000 questions and more than 200,000 answers.

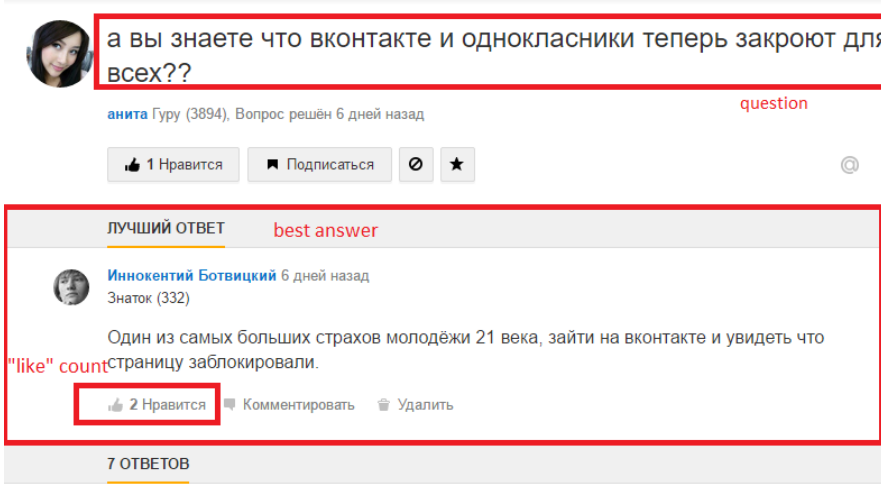

Fig. 1. Otvety@Mail.ru interface

[3] https://otvet.mail.ru/humor

4. Retrieval Models

As a baseline model we chose BM25 (Jones et. al, 2000) scoring, which is based on textual similarity between queries and documents. Stimuli in this model are mapped to lemmatized bag-of-words representations without stop words and then are queried against an inverted index.

One drawback BM25 has is that it requires word overlap between a query and a response, while some relevant responses may have no common words with the query. In the study we propose two models that address this issue: a word2vec-based semantic similarity model and a learning-to-rank approach using a diverse set of features. We also propose a Query Term Reweighting model, which is an enhancement of BM25 scoring.

4.1. Query Term Reweighting (QTR)

The proposed approach follows the idea of "humor anchors" introduced in (Yang, 2015). "Humor anchors" are words and phrases that are important for comic effect. Constituents of "humor anchor" may have low *idf* weights. For instance, the response presented in *Example 3* will not be ranked high enough by the baseline model because pronouns have low *idf* across the corpus, while the approach described below picks this as its top-1 response.

Example 3:
Question: Я прекрасно знаю, как с тобой разговаривать, не учи меня!
QTR: Ты разговариваешь со мной так, как будто у тебя есть абонемент в больницу
BM25: Разговаривать с единорогами.

Question: I know how to talk to you perfectly well, do not teach me!
QTR: You talk to me as if you have a seasonal ticket to a medical center
BM25: To talk with unicorns.

Firstly, we processed dialog jokes from the joke collection using a lemmatizer[4] (Korobov, 2015). To figure out what kinds of words are important for comic effect, we analyzed which morphological tags appear frequently in both questions and corresponding answers. In particular, we used a combination of part of speech and grammatical case. The most popular tags, without considering prepositions and conjunctions, were nominal pronouns, nouns in the nominative case, and verbs. Based on the acquired data, we composed a set of rules described below to adjust weights of anchor words using empirically derived boosting weights (see Table 2). These rules were applied to every stimulus before using BM25 weighting. All non-anchor words were excluded, and *tf-idf* weights of anchor words were multiplied by the corresponding boost values.

1. **Subjects.** Since there is a lack of accurate syntactic parsers for Russian, we defined a subject simply as a noun or a pronoun in the nominative case: a person, a place, a thing, or an idea that acts or is being described in a sentence ("Mother" in *Example 4*). The subject was appended to a query with the highest boost.

[4] https://github.com/kmike/pymorphy2#citing

Example 4:
Мама накричала на папу.
Mother shouted at dad.

2. **Named entities.** Words marked as proper names were also considered as main anchor words ("Russia" in *Example 5*). These words were added to the query with the same boost as subjects.

Example 5:
Как мы можем обустроить Россию?
How can we develop Russia?

3. **Question word context.** All nouns that were within three-word window with interrogative words (e.g. "who", "which", "when", etc.), like "alcohol" in *Example 6*, were added to the query with the highest boost.

Example 6:
Как исключить алкоголь?
How to give up alcohol?

4. **Anchor word context.** We extended the query with adjectives that were grammatically related to the subject ("best" in *Example 7*), as well as objects in a three-word window with the subject ("dad" in *Example 4*). An object is a noun, a noun phrase, or a pronoun that is affected by the action of a verb (a direct object or an indirect object) or that completes the meaning of a preposition (the object of a preposition).

Example 7:
Кто лучший тренер?
Who is the best coach?

5. **Verbs.** When the subject was found in a stimulus, we added verbs with a boost lesser than the boost of objects. Otherwise, verbs were appended with the highest boost. For instance, in *Example 8* the query will be extended by words "do", "get" and "pregnant".

Example 8:
— Что делать, чтобы не забеременеть?
— Моя девушка спит с другими парнями, чтобы не забеременеть от меня.

— What to do to not get pregnant?
— My girlfriend sleeps with other guys to not get pregnant by me.

6. **Pronouns.** For every first person or second person pronoun in the stimulus, we appended to the query an "opposite by person" pronoun with the highest boost. For instance, for the pronoun "I" the opposite one is "you", for "our"—"their", and so on. The original pronoun was appended to the query with a lesser boost. In *Example 3* the pronouns "I" and "you" and in *Example 9* "your" and "my" will be added to the query.

Example 9:

— Какое твое любимое блюдо?
— Мое любимое блюдо — макароны с сыром, потому что их название содержит рецепт и список ингредиентов одновременно.

— What is your favorite dish?
— My favorite dish is pasta with cheese, because its name contains a recipe and a list of ingredients at the same time.

Table 2. Empirically derived anchor boosts

Anchor Type	Boost
Subject	4.0
Named entity	4.0
Question word context	4.0
Inflected pronoun	4.0
Verb (no subject)	4.0
Anchor word context	3.0
Verb	2.5
Pronoun	1.5

4.2. Word2vec-Based Document Embeddings

The word2vec (Mikolov et al., 2013) method is a way to obtain word vectors such that semantically similar words have close vectors in terms of cosine similarity. There are also techniques to obtain document vectors of the same kind. One of them is doc2vec (Le and Mikolov, 2014)—a method that can infer vectors for new documents after training on thousands of sample documents. However, given a word2vec model, we can find a document vector just by the sum of vectors for the document words. (Lau and Baldwin, 2016) suggests that even though the sum-based representation is less effective than doc2vec, it often has better performance than bag-of-words (n-grams) in semantic-based tasks. Considering the lack of a publicly available doc2vec model for Russian and the comparable performance of the sum-based approach, we used the latter to obtain document vectors.

Specifically, we used a word2vec model trained on a Russian news corpus and provided by the service RusVectōrēs[5] (Kutuzov and Kuzmenko, 2017). We followed the same preprocessing as during the word2vec model construction—each text was mapped to a list of units in the form "lemma_POS" by Yandex Mystem 3.0[6] analyzer. We precalculated document vectors for our joke collection, and then, given a stimulus, we calculated its vector and found the closest jokes in terms of cosine similarity between vectors.

[5] http://rusvectores.org/en/about

[6] https://tech.yandex.ru/mystem/

4.3. Learning-to-Rank (LETOR)

Analogously to (Surdeanu et al., 2011), we used a learning-to-rank algorithm with a diverse set of features to re-rank responses of other models. In particular, we built a pool of answer candidates using top-50 answers returned by the BM25, QTR, and word2vec-based models described above. We used RankLib implementation of RankBoost algorithm to obtain a ranking function. The algorithm was trained on the CQA dataset, employing the following features for a question-answer pair.

1. Question length in characters.
2. Answer length in characters.
3. Question length in tokens.
4. Answer length in tokens.
5. BM25 score for the question-answer pair. As the score value is not bounded, we normalized it using score for the top-ranked document, hence obtaining a value between 0 and 1.
6. QTR model score for the question-answer pair. This score was normalized in the same fashion as the BM25 score.
7. word2vec-based model score for the question-answer pair.
8. IBM Model 1 probability. The IBM model 1 infers a word translation probability table from a parallel corpus. This table can then be used to estimate the probability of the answer being a translation of the question (Brown et al., 1993), which is known to perform well as a feature in question-answering ranking (Surdeanu et al., 2011). We trained this model on the CQA dataset, and then applied the same empirical trick as in (Surdeanu et al., 2011): probability of a word translating to itself was set to 0.5, and all other translation probabilities for the word were re-scaled to sum to 0.5.
9. Presence of an imperative verb. This is a binary feature that indicates whether for any verb in the question the same verb is present in the answer, but in the imperative mood.
10. Number of nouns, verbs, and adjectives in the answer that do not appear in the question. This feature is referred to as the "informativeness" of the answer (Surdeanu et al., 2011).
11. Similarity of POS-tag sequences of the question and the answer. Tags were obtained via pymorphy2[7] library, and similarity was calculated using the "gestalt pattern matching" algorithm (Ratcliff and Metzener, 1988).
12. Presence of rhyming words. This is a way to capture "puns" in the answers. To detect rhymes, we used Metaphone (Binstock and Rex, 1995) algorithm, specifically an implementation for Russian language—MetaphoneRU[8] library. Originally, the algorithm is used to find similar-sounding last names, but we used it to match nearly-rhyming words.

BM25, QTR, word2vec and translation probability scores, as expected, gave the highest ranking performance impact.

---

[7] https://github.com/kmike/pymorphy2

[8] https://github.com/Reaverart/MetaphoneRU

5. Evaluation

We evaluated the models in two ways: in the Humor[9] category of the Otvety@Mail.ru CQA platform and in laboratory settings.

5.1. CQA Platform

Perhaps the most important distinguishing feature of this evaluation method in comparison with other methods is that there are considerably more users. In average, there are about two new questions posted each minute in the Humor category of Otvety@Mail.ru.

We automatically posted top-1 ranked responses of each model for randomly sampled questions from this category during four days and gathered user reactions after a week. In total, bots answered 267 questions due to strict limitations of the CQA platform for user actions (30 answers per day for new account).

Table 3 provides the results obtained from Otvety@Mail.ru.

Table 3. User reactions from Otvet@Mail.ru (267 questions)

Model	Likes	# of best answers	Users that earned less likes	Best model
BM25	148	8	15.47%	14
QTR	142	16	14.91%	20
word2vec	147	14	15.82%	19
LETOR	156	12	16.93%	23
Oracle	197	50	24.67%	—

The "likes" column provides the total amount of "likes" for all answers of a model. The "best answer" column shows how many answers of the model were chosen as the best by question authors. The average percent of users who got less "likes" than the model in the same question thread is presented in the fourth column. Finally, the last column summarises how many times the model was better than the other ones. The model was considered as the "best" for a question if its answer was nominated as the "best answer" or earned more "likes" than answers of all other models. The last row of the table presents an "Oracle" model which chooses the most relevant answer within all mentioned models.

5.2. Lab Evaluation

Lab evaluation was conducted with the help of a dedicated annotation tool, see Fig. 2. Top-3 results for each model were selected for evaluation. Responses of all models were presented to an assessor in random order, three at a time. Responses were judged on a four-point scale (from 0 to 3, with the corresponding emoticons in the evaluation interface). We used pooling, each model was evaluated by four assessors independently. The assessors were instructed to pay close attention to question context during evaluation of the responses. As the test stimuli, we selected 80 questions from the ones we answered on the CQA platform.

[9] https://otvet.mail.ru/humor

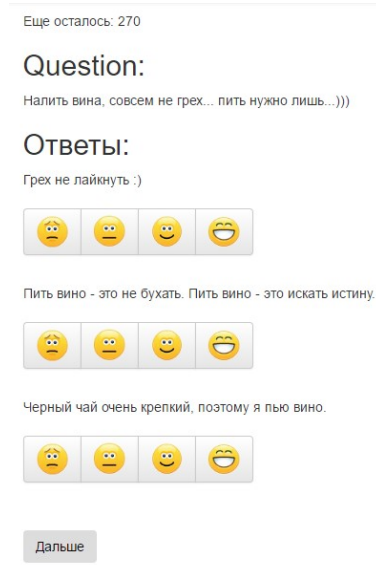

Fig. 2. The annotation tool for laboratory evaluation

Relevance score for a query–document pair is an average over all assessors' labels. Table 4 shows exemplary stimuli and the responses of the systems along with averaged assessors' judgments. We employed Discounted Cumulative Gain (DCG) (Järvelin and Kekäläinen, 2002) as the quality metric. Table 5 shows top-1 and DCG@3 scores for each model.

Table 4. Systems' responses and their evaluation scores

Score	Stimulus	Response
2.25	ВЫ меня опять стесняетесь?	Я рожден, чтобы стесняться незнакомой компании.
	Are YOU embarrassed by me again?	I was born to be embarrassed by unfamiliar company.
2.00	А если опоздать..., что будет? :)	Если опаздываешь, не торопись. Не надо опаздывать раньше времени.
	If you're late… what will happen? :)	If you're late, do not hurry. Do not be late ahead of time.
1.75	Налить вина, совсем не грех... пить нужно лишь...)))	Грехи снимают стресс.
	It's not a sin to pour some wine… just need to drink…)))	Sins relieve stress.
1.25	Никакие редуты не помогут... когда кролик атакует?)...	Недовольный кролик =)
	No strongholds will help... when a rabbit attacks?)...	Grumpy rabbit =)

Table 5. Lab evaluation results (80 questions)

Model	top-1	DCG@3
BM25	0,76	1,48
QTR	0,85	1,58
word2vec	0,77	1,62
LETOR	0,74	1,41
Oracle	1,63	2,95

We also calculated Cohen's kappa (Carletta, 1996) as a measure of inter-annotator agreement. We used weighted variant (weights are absolute differences between labels) for pair-wise agreement. Averaged pairwise kappa statistics for four assessors in our experiments is 0.21. Example 10 illustrates QA-pair with low assessor agreement (assessor 1—☹, assessor 2—😁, assessor 3—☹, assessor 4—☺).

Example 10:

— Мысли разбежались) Как собрать—чтоб не повредить—мысли?
— «Далеко пойдешь!» — подумала мысль... и ушла.

— Thoughts have dispersed) How can they be gathered without damage?
— "You'll go far!" a thought reflected… and went away.

6. Discussion and Future Work

The results of the evaluation on the CQA platform show that the learning-to-rank approach, which was trained on the historical CQA dataset, provides the best performance. In particular, as shown in Table 3, the LETOR model is ahead of other models in terms of "likes" and "best model" measures. Moreover, it provides answers that on average have more likes than around 17% of answers provided by users of the CQA platform.

On the other hand, QTR approach has the biggest amount of "best answers" and the least amount of "likes" at the same time. The word2vec-based approach has comparable performance. We noticed that answers marked as the "best" on average have less "like" marks than other answers. This suggests that askers often disagree with the community about which answers are appropriate or funny.

The most surprising aspect of the manual evaluation is that the LETOR method shows the lowest value in both top-1 and DCG@3 metrics. There are two possible explanations for this. The first one is based on the low inter-annotator agreement. Such a low agreement confirms that the perception of humor varies greatly from person to person, and conclusive lab evaluation may require a significantly higher number of assessors. Yet another explanation of the drastic drop in the LETOR performance is that some CQA users positively evaluate answers that are not quite in the context, and thus training on the CQA data can yield a biased model. This hypothesis can be investigated in future studies by training the LETOR model using the QA pairs evaluated in laboratory settings.

The findings suggest that information retrieval approach is a promising direction in humorous response generation. It is also clear that morphological and word2vec-based features are effective for the task. Nevertheless, the results of the "oracle" model

indicate that there is an abundant room for the improvement of the answer ranking. Thus, in future investigations we plan to enhance the learning-to-rank approach by incorporating features that can capture the nature of humor and context in a better way.

References

1. *Binstock A., Rex J.* (1995), Practical algorithms for programmers, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA
2. *Blinov V., Bolotova V., Braslavski P.* (2016) Information retrieval approach to humorous response generation in dialog systems: a baseline, available at: http://www.dialog-21.ru/media/3462/blinov.pdf
3. *Brown P. F., Pietra V. J. D., Pietra S. A. D., Mercer R. L.* (1993), The mathematics of statistical machine translation: Parameter estimation, Computational linguistics, Vol. 19(2), pp. 263–311.
4. *Carletta J.* (1996), Assessing agreement on classification tasks: the kappa statistic, Computational linguistics, Vol. 22(2), pp. 249–254.
5. *Friedland L., Allan J.* (2008), Joke retrieval: recognizing the same joke told differently, Proceedings of the 17th ACM conference on Information and knowledge management, Napa Valley, California, USA, pp. 883–892.
6. *Hong B. A., Ong E.* (2009), Automatically extracting word relationships as templates for pun generation, Proceedings of the Workshop on Computational Approaches to Linguistic Creativity, pp. 24–31.
7. *Jones K. S., Walker S., & Robertson S. E.* (2000), A probabilistic model of information retrieval: development and comparative experiments: Part 2, Information processing & management, Vol. 36(6), pp. 809–840.
8. *Järvelin K., Kekäläinen J.* (2002), Cumulated gain-based evaluation of IR techniques, ACM Transactions on Information Systems, Vol. 20(4), pp. 422–446.
9. *Khooshabeh P., McCall C., Gandhe S., Gratch J., Blascovich J.* (2011), Does it matter if a computer jokes, CHI '11 Extended Abstracts on Human Factors in Computing Systems, Vancouver, BC, Canada, pp. 77–86.
10. *Korobov M.* (2015), Morphological analyzer and generator for Russian and Ukrainian languages, Analysis of Images, Social Networks and Texts. Communications in Computer and Information Science, Vol. 542, Springer, Cham, pp. 320–332.
11. *Kutuzov A., Kuzmenko E.* (2017), WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models, Analysis of Images, Social Networks and Texts. AIST 2016. Communications in Computer and Information Science, Vol. 661. Springer, Cham, pp. 155–161.
12. *Lau J. H., Baldwin T.* (2016), An empirical evaluation of doc2vec with practical insights into document embedding generation, Proceedings of the 1st Workshop on Representation Learning for NLP, Berlin, Germany, pp. 78–86.
13. *Le Q. V., Mikolov T.* (2014), Distributed Representations of Sentences and Documents, Proceedings of The 31st International Conference on Machine Learning, Beijing, China, pp. 1188–1196.
14. *Mihalcea R., Strapparava C.* (2006), Technologies that make you smile: Adding humor to text-based applications, IEEE Intelligent Systems, Vol. 21(5), pp. 33–39.

15. *Mikolov T., Chen K., Corrado G., Dean J.* (2013), Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.
16. *Nijholt A.* (2002), The April Fools' Day Workshop of Computational Humour, Trento, Italy, pp. 101–111.
17. *Ratcliff J., Metzener D.* (1988), Pattern matching: The Gestalt approach, Dr. Dobb's Journal, p. 46.
18. *Ritchie G.* (2005), Computational mechanisms for pun generation, Proceedings of the 10th European Natural Language Generation Workshop, Aberdeen, Scotland, UK, pp. 125–132.
19. *Ritter A., Cherry C., Dolan W. B.* (2011), Data-driven response generation in social media, Proceedings of the conference on empirical methods in natural language processing, Edinburgh, Scotland, UK, pp. 583–593.
20. *Stock O., Strapparava C.* (2003), Getting serious about the development of computational humor, Proceedings of the 18th international joint conference on Artificial intelligence, Acapulco, Mexico, pp. 59–64.
21. *Surdeanu M., Ciaramita M., Zaragoza H.* (2011), Learning to rank answers to non-factoid questions from web collections, Computational linguistics, Vol. 37(2), pp. 351–383.
22. *Valitutti A., Toivonen H., Doucet A., Toivanen J. M.* (2013), "Let Everything Turn Well in Your Wife": Generation of Adult Humor Using Lexical Constraints, Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, pp. 243–248.
23. *Yang D., Lavie A., Dyer C., Hovy E.* (2015), Humor Recognition and Humor Anchor Extraction, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp. 2367–2376.