Smart Context Generation for Disambiguation to Wikipedia

Andrey Sysoev[1(✉)] and Irina Nikishina[1,2]

[1] Ivannikov Institute for System Programming, Russian Academy of Sciences, Moscow, Russia
{sysoev,nia}@ispras.ru
[2] Higher School of Economics, Moscow, Russia

Abstract. Wikification is a crucial NLP task that aims to identify entities in text and disambiguate their meaning. Being partially solved for English, the problem still remains fairly untouched for Russian. In this article we present a novel approach to Disambiguation to Wikipedia applied to the Russian language. Inspired by the Neural Machine Translation task our method implements encoder-decoder neural network architecture. It translates text tokens into concept embeddings that are subsequently used as context for disambiguation. In order to test our hypothesis we add our context features to GLOW system considered a baseline. Moreover, we present commonly available dataset for the Disambiguation to Wikipedia task.

Keywords: Disambiguation to Wikipedia · Wikification for Russian
Encoder-decoder neural network architecture · Concept embeddings

1 Introduction

It is widely acknowledged that Wikipedia has almost become the most popular and authoritative source in the modern Internet society, remaining the largest multi-language corpus that is especially useful for different NLP tasks. In particular, Wikipedia might be useful for Named Entity recognition [17,20], word sense disambiguation [5], text classification [13] and other tasks that require additional information about real world enitites that could be gained by means of Wikification.

Wikification task consists of two levels: one is responsible for locating entities in raw text, the other stands for associating entities with the appropriate Wikipedia pages – hereinafter concepts. The last step is also known as Disambiguation to Wikipedia (D2W) and might also be considered a separate task: to each mention m assign a Wikipedia concept e or a special nil value (not-yet-in-Wikipedia concept). For instance, "St. Petersburg" in sentence "First time I saw St. Petersburg last year" may refer either to the Russian city or to the city in the United States or even to the Iranian comedy film. The goal of a D2W

D. Ustalov et al. (Eds.): AINL 2018, CCIS 930, pp. 11–22, 2018.
https://doi.org/10.1007/978-3-030-01204-5_2

system in this case is to associate "St. Petersburg" with the correct Wikipedia concept.

While most papers about Wikification and D2W describe new methods implemented for English or other European languages, very little research is made for Russian. That is why in the current paper we present a novel approach to D2W in application to the Russian language.

We assume that context used for disambiguation may also be generated with the help of Neural Machine Translation (NMT) techniques. Thus our idea is to build a system that transforms text tokens into a set of concept embeddings – smart context.

According to our hypothesis of "token-to-concept" translation, sentence "She ate too much Caesar at Gordon Ramsay yesterday" should be translated to the language of concepts as "Caesar salad, Restaurant Gordon Ramsay" and not "Julius Caesar, Gordon Ramsay". We expect concept embeddings generated by NMT model act as appropriate unambiguous context for the D2W task.

In order to evaluate usefulness of proposed features, based on similarity to generated smart context, we implement the approach from [18] as the baseline. We also create a dataset for the Russian language, which is described in Sect. 5.1.

Therefore, the main contribution of our research is the following: we apply the existing D2W method to Russian, demonstrate the advantages of the developed smart context based features and propose the generated dataset as the gold standard dataset for the D2W task.

2 Related Work

Our approach is based on application of encoder-decoder architecture borrowed from NMT research area to solve D2W problem. That is why we suppose being important to review related work in both fields.

2.1 Wikification and Disambiguation

As a subtask of Entity Linking, Wikification for the English language has quite a long history. The whole timeline is perfectly described in [22], while we draw our attention to those works which are more important for our research.

First, two prominent studies for Wikification and D2W tasks should be mentioned: [16] where standard measures like commonness and relatedness are proposed and [18] that introduces Global and Local algorithms for Entity Disambiguation (GLOW). GLOW system from the second paper is also described in Sect. 3 in more detail.

Furthermore, we should mention [6] as it proves Entity Linking to be quite useful for other NLP tasks. In [7] it is also demonstrated that capturing topic at multiple granularities from text via a CNN model is essential for concept disambiguation.

Besides our research the idea of generating concept embeddings is also developed in [8]. Concept vectors are trained there using word2vec [15] and then

utilized for generating local context attention. For global disambiguation they propose using Conditional Random Fields and Loopy Belief Propagation. The authors compare their approach to and mostly outperform [3] and [11].

One of the most recent works about D2W is [23], in which authors apply the Random Forest algorithm for mention disambiguation. To decide whether an entity should be included to the result set they use helpfulness evaluation based on link probability, entity popularity, entity class and topical coherence.

Concerning D2W for the Russian language, we suppose that its current state is only a starting point. Besides [21] who try to implement maximum entropy classifier likewise in [16] for Russian and test in on private corpus, we are not aware of other works devoted to the current topic.

2.2 Neural Machine Translation

With the recent developments in Deep Neural Networks, NMT is closely associated with sequence-to-sequence model [19]. This approach generally comprises two stages: *encoding stage* that converts sentence from source language into a vector representing its language-independent meaning and *decoding stage*, responsible for translating this vector into sentence written in target language.

A few years ago NMT systems like [4] and [24] implemented bidirectional Long Short-Term Memory (LSTM) models for both encoding and decoding phases. Later [9] integrated Convolutional Neural Networks (CNN), applying convolutional model instead of LSTM to encoder and then even to decoder [10]. In the current study we are comparing biLSTM and CNN based encoders (Sects. 4.2 and 4.3) with regards to the D2W task.

Another constituent part of NMT model is the attention mechanism that allows to learn alignments between source and target sentences. For the first time it is used in [1], then improved by [14]. Application of attention weights for the current research might be rather unevident and is thoroughly described in Sect. 4.4.

3 Baseline

As a baseline solution for D2W task we select GLOW approach from [18]. In this section we briefly describe the algorithm itself along with our modifications and clarifications.

GLOW starts with enriching provided collection $\{m_1, m_2, \ldots, m_N\}$ with extra mentions, computed as named entities and noun phrases of length not more than 5. Each mention m is associated with its possible meanings E_m, extracted from Wikipedia redirects and anchor texts; in correspondence to [18], only top 20 most frequent concepts are analysed.

Then come two main GLOW phases: first of all, global context, which consists of a number of input text describing concepts, is identified; secondly, this context is used to determine the final assignment of concepts to input mentions. Each

phase is based on ranker-linker pair of machine learning algorithms which differ only in the set of features being used.

Ranker accepts a mention m with possible meanings E_m within a document D and grades all E_m according to their plausibility of being correct disambiguation of m. Ranker training is performed with RankSVM [12].

For each mention m **linker** is provided with ranker-computed scores of possible meanings; its goal is to filter out presumably incorrect assignments, when ranker fails to deliver the highest weight to the correct meaning. Linker is a conventional binary linear SVM classifier.

During context generation phase ranker uses context independent and local context features. For computing final meaning-concept assignment global context features are used as well.

3.1 Context Independent Features

Context independent features include $P(e|m)$ and $P(e)$. $P(e|m)$ – commonness – indicates how often mention m links to entity e in Wikipedia. $P(e)$ is the portion of Wikipedia articles, which have links to e.

3.2 Local Context Features

Let us introduce the following notations: $text(m)$ is TFIDF vector of document D, containing m; $context(m)$ is TFIDF vector computed for w-size window around m; $text(e)$ contains $2w$ elements with top TFIDF weight extracted from Wikipedia page corresponding to concept e; $context(e)$ is similar to $text(e)$ but is collected through all w-size windows around mentions, linked to e throughout the whole Wikipedia. In contrast to [18] we utilize lemmas instead of tokens to gain $text(\cdot)$ and $context(\cdot)$.

Local context features include cosine similarity calculated for the following pairs of vectors:

$text(m) \leftrightarrow text(e)$,
$text(m) \leftrightarrow context(e)$,
$context(m) \leftrightarrow text(e)$,
$context(m) \leftrightarrow context(e)$.

Additionally, [18] uses reweighted versions of described features, which are aimed at changing token importance in TFIDF vectors: boost more specific and fine less specific tokens for the given possible meanings of the mention m. Reweighted TFIDF is evaluated according to the formula:

$$w_{text}(l, e, m) = \frac{text(e)_{[l]}}{\sum_{e' \in E_m} text(e')_{[l]}}, \tag{1}$$

where $text(e)_{[l]}$ is weight of lemma l in TFIDF vector $text(e)$, which is assumed to be 0 if vector $text(e)$ does not contain l. $w_{context}(l, e, m)$ is computed similarly.

3.3 Global Context Features

Let us introduce some notations (see Table 1).

Table 1. Notations for computing global context features.

agg_G	***Aggregating function***
max_G	Maximum value computed throughout the whole global context G
avg_G	Average value computed throughout the whole global context G
$\mathbb{1}$	***Concepts link indicator***
$\mathbb{1}_{e_i - e_j}$	Binary indicator of e_i having a link to e_j *or* vice versa
$\mathbb{1}_{e_i \leftrightarrow e_j}$	Binary indicator of e_i having a link to e_j *and* vice versa
sim	***Similarity***
PMI'	Pointwise Mutual Information similarity measure (formulas 2 and 3)
NGD	Normalized Google Distance (formula 4)
$links$	***Link set***
in_links	Set of concepts, which have an outgoing link to e
out_links	Set of concepts, to which e has an outgoing link

$$PMI' = \frac{PMI}{1 + PMI}, \tag{2}$$

$$PMI(L_1, L_2) = \frac{|E||L_1 \cap L_2|}{|L_1||L_2|}, \tag{3}$$

$$NGD(L_1, L_2) = \frac{log\ max(|L_1|, |L_2|) - log|L_1 \cap L_2|}{log|E| - log\ min(|L_1|, |L_2|)}, \tag{4}$$

where E is a set of all Wikipedia concepts.

Global features are constructed by composing combinations of introduced options, peeking one at a time: $agg_{g \in G}\ \mathbb{1} \cdot sim(links(e), links(g))$. For instance, a sample feature $F(e)$ is $\max_{g \in G} \mathbb{1}_{e-g} \cdot PMI'(in\_links(e), in\_links(g))$.

A pair of extra global features utilized in GLOW is $max_{g \in G}\ \mathbb{1}_{e \leftrightarrow g}$ and $avg_{g \in G}\ \mathbb{1}_{e \leftrightarrow g}$.

3.4 Linker Features

Linker features include the same set of features as its corresponding ranker. However, there is a number of additional features:

- difference in score between the best and the second-best concept, produced by ranker;
- entropy of possible mention meanings;
- indicator of meaning being a named entity;

- the fraction of mention appearances in Wikipedia, where it is used as a link;
- Good-Turing estimate of mention not having correct meaning described in Wikipedia. We use the following formula:

$$F_{GoodTuring}(m) = \frac{\sum_{e \in E_m} \mathbb{1}_{count(e,m)=1}}{\sum_{e \in E_m} count(e, m)}, \tag{5}$$

where $count(e, m)$ is the number of times mention m is linked to concept e in Wikipedia.

4 Similarity to Generated Context

In this section we introduce a novel type of context, which is exploited in D2W. Additionally, we propose a method for computing similarities from concept to generated context, which are used as extra features in GLOW algorithm.

4.1 Context Generation

The proposed type of context – smart context – is the result of translating input text into a "language of concepts" with some neural machine translation approach. In this work we utilize a simple encoder-decoder architecture, proposed in [19].

Input tokens (with special `END_TOKEN` appended) are mapped into their embeddings and then fed into encoder part of the network. Encoder translates them into internal representation I, which is further passed to decoder. Additionally, encoder compresses the whole input into a pair of fixed-length vectors (c, h), which are later used in decoder initialization (see Fig. 1).

Decoder part of neural network is based on LSTM. Encoder's (c, h) pair is passed through fully-connected layers F_c and F_h to match decoder LSTM state size and is used to initialize it. Tokens internal representation I is aggregated in conformance to attention mechanism [1] and further fed into decoder LSTM. Moreover, each LSTM cell also consumes decoder output from the previous step (initially, special `START_CONCEPT` is passed instead). Each LSTM cell output and newly computed attention vector are passed through fully-connected layer F_s, L2-normalized and then considered a target context concept embedding. Decoding stops when special `END_CONCEPT` embedding is produced.

We experiment with two types of encoder architectures – biLSTM-based and CNN-based, which are described in detail in Sects. 4.2 and 4.3 correspondingly.

4.2 BiLSTM-Based Encoder

Token embeddings are passed as input to biLSTM (each LSTM is of size l), which converts them into internal representation I. Hidden state vectors $\overrightarrow{c}$ and $\overleftarrow{c}$ of forward and backward LSTMs are concatenated to form final vector c. (h is computed in the same way). To introduce regularization to our model, dropout layers with keeping probability p are applied to I, c and h before returning them from encoder (see Fig. 2).

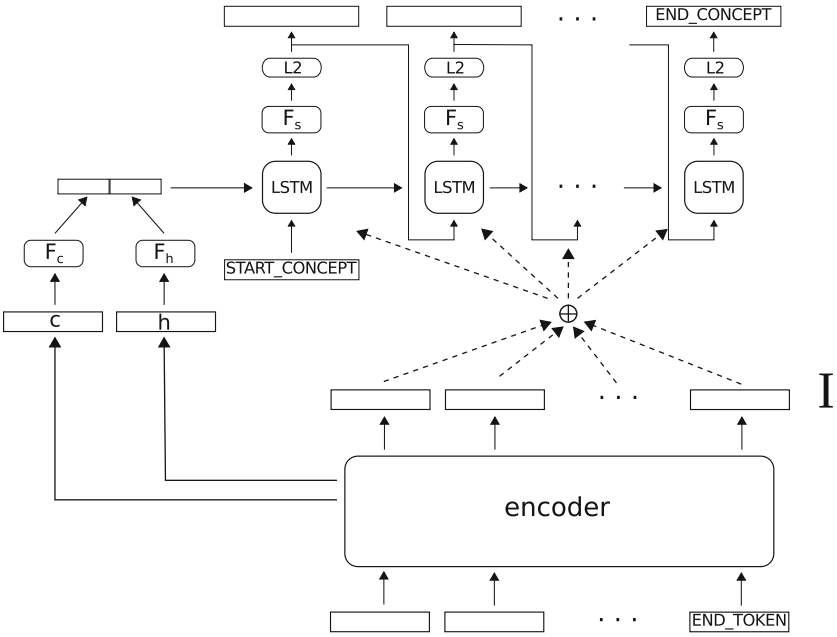


Fig. 1. Encoder-decoder neural network architecture.

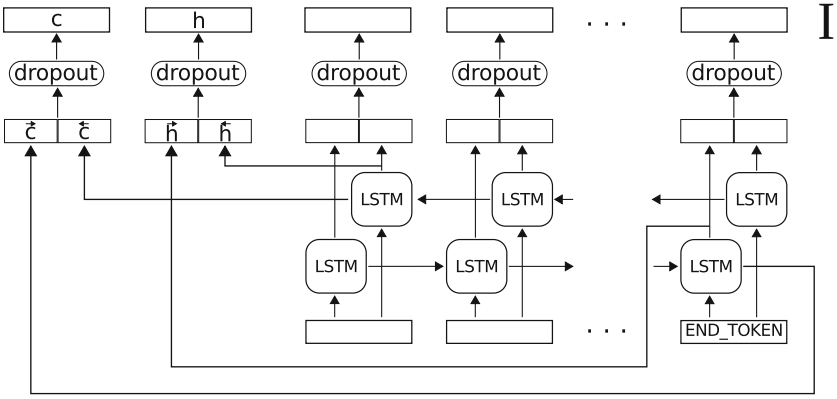
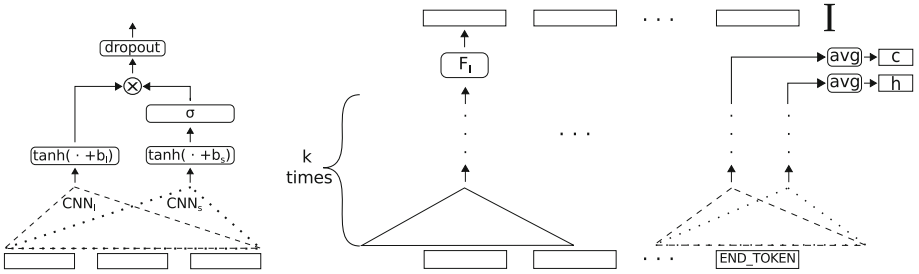


Fig. 2. BiLSTM-based encoder.



(a) Gated unit. (b) Application of gated unit in encoder.

Fig. 3. CNN-based encoder.

4.3 CNN-Based Encoder

CNN-based encoder architecture is hugely inspired by [10]; it mainly consists of several CNN-based gated units (see Fig. 3a):

$$g(T) = v_l(T) \otimes \sigma(v_s(T)), \tag{6}$$

$$v.(T) = tanh(cnn.(T) + b.), \tag{7}$$

where T is a matrix of token embeddings, $\otimes$ is the pointwise vector multiplication, $cnn(T)$ is the result of application of CNN layer to matrix T, b is a trainable variable. Output of each unit is then passed through dropout layer with keeping probability p.

CNN-based gated units are stacked one upon another for k times. Output of the final block is averaged; it constitutes the final c vector. Vector h is computed in the same way, but using a separate stack of blocks. Another stack is used to compute internal representations I, but its final block output is preliminary traversed through fully-connected layer F_I (see Fig. 3b).

4.4 Similarity to Smart Context Computation

Similarity features $F_S^{\cdot}(e, m)$ from concept e to smart context S for mention m are computed according to the following formulas:

$$F_S^{max}(e, m) = \max_{s \in S} cos(embedding(e), s), \tag{8}$$

$$F_S^{avg}(e, m) = \frac{1}{|S|} \sum_{s \in S} cos(embedding(e), s), \tag{9}$$

$$F_S^{attention_{max}}(e, m) = \max_{s \in S} \frac{\alpha(s, m)}{\sum_{s \in S} \alpha(s, m)} cos(embedding(e), s), \tag{10}$$

$$F_S^{attention_{avg}}(e, m) = \frac{1}{\sum_{s \in S} \alpha(s, m)} \sum_{s \in S} \alpha(s, m) cos(embedding(e), s), \tag{11}$$

$$\alpha(s, m) = \sum_{t:\ t\ intersects\ m} attention(t, s), \tag{12}$$

where t is text token, $attention(t, m)$ is decoder attention for token t when computing concept embedding s. In other words, to compute each context embedding weight $\alpha(s, m)$ we sum attention scores of mention tokens, returned by decoder.

5 Evaluation

In the current section we describe dataset prepared for training and testing GLOW and neural network parameters. Furthermore, we evaluate the results obtained from the algorithms described above.

5.1 Data and Parameters

While for the English language there exists a large amount of corpora for the disambiguation task, there is no open dataset available for Russian. Thus, we download the Russian Wikipedia dump of May 1, 2018 that contains more than 1470000 articles and build our own corpus. We collect those pages that attain one of the two best grades in WikiProject article quality evaluation scheme: we select 2968 articles from labelled as `Good article` for training; 1056 articles categorized as `Featured article` are treated as test set[1]. For training our neural network models we omit `Featured articles` in order to avoid possible overlapping with test data.

Moreover, the Wikipedia dump is utilized for fitting embedding models. We pre-train a word2vec [15] model (size = 100, window = 5, skip-gram) for generating concept embeddings and a fasttext [2] model (size = 100, window = 5, skip-gram) for tokens.[2]

Table 2 outlines neural network hyperparameter values used during the experiments.

5.2 Evaluation Results

In this section we explore the usefulness of our features, based on concept similarity to smart context, on the D2W task.

In order to carry out fair evaluation we calculate the minimum level for the results which is known as `Most common sense` (MCS in Table 3). For each mention the model selects the most popular meaning (if several), according to its commonness value. `Upper bound` is an oracle, which always predicts correct meaning if it is *nil* or is among top 20 most frequent mention meanings. GLOW-based methods select meanings from the same set, thus `upper bound` shows the best quality our approach may achieve.

[1] https://github.com/ispras-texterra/ainl-2018-d2w-dataset.

[2] Note, that token embedding size is 101 = 100+ extra position to encode `END_TOKEN`. Similar idea is for concept embedding size and `START_CONCEPT`/`END_CONCEPT`.

Table 2. Hyperparameters.

Section	Parameter	Label	Value
3.2	Window size around mentions	w	100
4.1	Token embedding size		101
	Concept embedding size		102
	Fully-connected layers F_c, F_h size		500
	Decoder LSTM size		500
	Fully-connected layer F_s size		102
	Attention		Bahdanau [1]
	Attention size		500
	Loss function		cosine distance
	Optimizer		Nadam
	Batch size		16
4.2	Forward/backward LSTM size	l	500
	Number of epochs		1110
4.3	Fully-connected layer F_I size		500
	Size of gated units stack	k	3
	CNN filter size		5
	Number of CNN filters		500
	Gated unit bias b size		500
	Number of epochs		2131
4.2 and 4.3	Keeping probability (train)	p	0.7

Implementation of GLOW system for Russian is fairly significant, as the results for this model outperform MCS by more than 4 percentage points. We additionally found out that GLOW without linker (which simply accepts top-scored concepts returned by ranker) performs even better. Applying features based on CNN and biLSTM generated smart context further improves the results.

Table 3. Evaluation results.

Model	Macro-averaged accuracy, %
MCS	83.01
Upper bound	94.03
GLOW	87.80
GLOW$_{no_linker}$	88.01
GLOW + smart context features (CNN)	88.04
GLOW$_{no_linker}$ + smart context features (CNN)	88.25
GLOW + smart context features (biLSTM)	88.19
GLOW$_{no_linker}$ + smart context features (biLSTM)	**88.30**

To prove usefulness of the proposed features we split test data into 10 parts and evaluate baselines (GLOW and GLOW_{no_linker}) and our best approach (GLOW_{no_linker} with biLSTM-based smart context features) on each part separately. Application of Wilcoxon signed rank test shows that our approach is better than baselines and the results are statistically significant with p-value <0.002.

6 Conclusion

In the current paper we propose a novel approach for generating context for the D2W task. Our method implies encoder-decoder architecture for translating sequence of tokens into concept embeddings.

During the research we trained two models with CNN and biLSTM based encoders and then compared their performance with the GLOW approach [18] implemented as baseline. Both of them outperform GLOW.

Despite moderate quality improvement of the result for the D2W task, we still consider the idea of translating tokens into concepts legible and expect to implement the encoder-decoder approach not only for D2W but for the whole Wikification task. Another line of work is to evaluate our approach on standard datasets for the English language.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of 3rd International Conference for Learning Representations, San Diego, pp. 1–15 (2015)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017)
3. Cheng, X., Roth, D.: Relational inference for wikification. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1787–1796 (2013)
4. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: encoder-decoder approaches. In: Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8) (2014)
5. Dandala, B., Mihalcea, R., Bunescu, R.: Word sense disambiguation using wikipedia. In: Gurevych, I., Kim, J. (eds.) The People's Web Meets NLP, pp. 241–262. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35085-6_9
6. Durrett, G., Klein, D.: A joint model for entity analysis: coreference, typing, and linking. Trans. Assoc. Comput. Linguist. **2**, 477–490 (2014)
7. Francis-Landau, M., Durrett, G., Klein, D.: Capturing semantic similarity for entity linking with convolutional neural networks. In: Proceedings of NAACL-HLT, pp. 1256–1261 (2016)
8. Ganea, O.E., Hofmann, T.: Deep joint entity disambiguation with local neural attention (EMNLP 2017). In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2619–2629. Association for Computational Linguistics (2017)

9. Gehring, J., Auli, M., Grangier, D., Dauphin, Y.: A convolutional encoder model for neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics: Long Papers, vol. 1, pp. 123–135 (2017)
10. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: International Conference on Machine Learning, pp. 1243–1252 (2017)
11. Guo, Z., Barbosa, D.: Robust named entity disambiguation with random walks. Semant. Web 1–21 (2016, preprint)
12. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 133–142. ACM (2002)
13. Li, J., Cai, Y., Cai, Z., Leung, H., Yang, K.: Wikipedia based short text classification method. In: Bao, Z., Trajcevski, G., Chang, L., Hua, W. (eds.) DASFAA 2017. LNCS, vol. 10179, pp. 275–286. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-55705-2_22
14. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421 (2015)
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
16. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 509–518. ACM (2008)
17. Nothman, J., Ringland, N., Radford, W., Murphy, T., Curran, J.R.: Learning multilingual named entity recognition from Wikipedia. Artif. Intell. **194**, 151–175 (2013)
18. Ratinov, L., Roth, D., Downey, D., Anderson, M.: Local and global algorithms for disambiguation to Wikipedia. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 1375–1384. Association for Computational Linguistics (2011)
19. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
20. Sysoev, A., Andrianov, I.: Named entity recognition in Russian: the power of wiki-based approach. In: Proceedings of International Conference "Dialogue-2016", pp. 746–755 (2016)
21. Turdakov, D., et al.: Semantic analysis of texts using texterra system (2014). http://www.dialog-21.ru/digests/dialog2014/materials/pdf/TurdakovDY.pdf. Accessed 28 May 2018
22. Wu, G., He, Y., Hu, X.: Entity linking: an issue to extract corresponding entity with knowledge base. IEEE Access **6**, 6220–6231 (2018)
23. Yamada, I., Ito, T., Takeda, H., Takefuji, Y.: Linkify: enhancing text reading experience by detecting and linking helpful entities to users. IEEE Intell. Syst. (2018)
24. Zhou, J., Cao, Y., Wang, X., Li, P., Xu, W.: Deep recurrent models with fast-forward connections for neural machine translation. Trans. Assoc. Comput. Linguist. **4**(1), 371–383 (2016)