Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2016”

Moscow, June 1–4, 2016

РАСПОЗНАВАНИЕ ИМЕНОВАННЫХ СУЩНОСТЕЙ: ПОДХОД НА ОСНОВЕ ВИКИ-РЕСУРСОВ

Сысоев А. А. (sysoev@ispras.ru),
Андрианов И. А. (ivan.andrianov@ispras.ru)

Институт системного программирования
РАН, Москва, Россия

Ключевые слова: распознавание именованных сущностей, вики-ресурсы, машинное обучение, векторное представление слов

NAMED ENTITY RECOGNITION IN RUSSIAN: THE POWER OF WIKI-BASED APPROACH

Sysoev A. A. (sysoev@ispras.ru),
Andrianov I. A. (ivan.andrianov@ispras.ru)

Institute for System Programming of RAS, Moscow, Russia

Named entity recognition and classification is an important natural language processing task, aimed at finding words and word sequences, which denote named entities of different types in plain texts. This challenge was addressed in Task 1 of FactRuEval-2016 evaluation.
In the context of this evaluation, our team, acting for the Institute for System Programming of the Russian Academy of Sciences, proposed two approaches to exploiting information, mined from Wikidata and Wikipedia, for improving quality of named entity detection methods. In the first approach word2vec word embeddings, computed on Wikipedia, are used along with basic features in tokens classification. The second approach utilizes both Wikipedia and Wikidata to automatically construct a representative corpus for named entity recognition and classification training. Additionally, Wikidata, treated as a property graph, is used to collect named entity specific word dictionaries.

Our approaches (marked with identifier ‘Orange’ in FactRuEval-2016 organizers’ quality evaluation reports) show up promising results, doing especially good for such well-defined class as person, still being appropriate for detecting named entities of other types as well.

Key words: named entity recognition, wiki, machine learning, word embedding, word2vec

Introduction

Named entity recognition and classification (NERC) is an important information extraction task which is aimed at sifting through plain texts for such information units as human names, locations, organizations, facilities, products, dates, geopolitical entities, holidays and so on.

During FactRuEval-2016 evaluation, Task 1 was fully devoted to classical NERC. Participants’ efforts were to be concentrated on detecting entities of three most popular types—person, location, organization. Additionally, some sophistication of the problem was introduced: it was required to distinguish between locations, meaning some geographical place, and locations, meaning an organization or group of people, as in ‘Russia is celebrating Victory Day’. The later entity type is referred to as locorg.

Results of our team, acting for Information Systems Group of the Institute for System Programming of the Russian Academy of Sciences, are marked with identifier ‘Orange’ in FactRuEval-2016 final evaluation reports.

The rest of the article is structured as follows. Related work is discussed in Section 1. Section 2 outlines some important features of wiki-resources, exploited in our work. Section 3 describes our approach to FactRuEval-2016 named entity recognition task. In Section 4 evaluation results are provided. We wrap up with some concluding thoughts in the final section.

1. Related work

One of the earliest approaches to named entity recognition and classification was based on human-defined rules and heuristics (Rau, 1991). For now methods, based on machine learning, seem to be more promising (Zhang & Johnson, 2003), as they are easier to build and adapt to new domains. But they still face a major natural language processing problem—data sparsity: words are represented with numerical vectors of high dimensionality, thus requiring huge train corpus for building a representative model. Initial attempt to cope with data sparsity was building dictionaries of somehow similar words and adding features, indicating whether classified word belongs to one of them (Zhang & Johnson, 2003). Recent approaches utilize language models (Miller et al., 2004)—Brown clusters (Brown et al., 1992), word2vec (Mikolov et al., 2013)—in attempt to reduce problem dimensionality. Another way is to semi-automatically construct more representative training corpus (Nothman et. al, 2013).

In our work we try to compare both approaches: in the first arrangement we bet on word2vec features, computed on Wikipedia; in the second arrangement we stake on training upon automatically created corpus, made up from Wikipedia and Wikidata.

2. Wiki-resources overview

For better understanding of our approach, some Wikipedia and Wikidata features, vital for our work, are briefly described below.

2.1. Wikipedia

Wikipedia (www.wikipedia.org) is a free online encyclopedia, which can be updated and edited by any user. Its large volume and rich link structure make Wikipedia an invaluable resource for solving many natural language processing problems (Milne & Witten, 2008; Ratinov et al., 2011; Turdakov et al., 2014). Wikipedia's textual articles describe entities of the real world, linked with each other to simplify navigation through different parts of the encyclopedia.

2.2. Wikidata

Wikidata (www.wikidata.org) is a free online machine-readable multilingual knowledge base, interlinked with other Wiki resources. Wikidata can be thought of as a property graph (Jouili & Vansteenberghe, 2013), where ontological classes and instances are represented with vertices and relations—with edges. In Wikidata terminology they are called, respectively, items and properties. Items are assigned special identifiers, started with 'Q' and followed by some number; property identifiers start with 'P', followed by some number as well. Both items and properties may have several textual representations on the specified language; some representation is selected as the main one, called label, while others are called aliases. Items may also have links to corresponding Wikipedia articles on different languages.

For example, Wikidata contains such items as 'geographical object' (Q618123, Wikipedia article: 'Geographical feature'), 'fictional location' (Q3895768, Wikipedia article: 'Fictional location', aliases: 'fictional place', 'mythical location', 'legendary place', ...), 'Narnia' (Q2886622, Wikipedia article: 'Narnia (country)'). Sample properties are: 'subclass of' (P279), 'instance of' (P31), 'given name' (P735). One can infer relations between unconnected items with Wikidata graph traversal: 'Narnia' (Q2886622) is an instance of 'fictional location' (Q3895768), as there is an 'instance of'-'subclass of' path through 'fictional country or state' (Q1145276).

3. Method description

Our method is based on sequential traversal and classification of text tokens. Computed token labels are then used to determine named entity type and boundaries in text.

Valid labels are constructed from supported named entity types using BILOU encoding scheme (Uchimoto et al., 2000). Four labels are generated for each type: B-TYPE (for example, B-PERSON, B-ORGANIZATION) is used to indicate entity beginning, I-TYPE indicates token in the middle of the entity, L-TYPE specifies ending of the entity, U-TYPE marks single token entities. Additionally, there is O label, which is assigned to non-entity tokens. In FactRuEval-2016 evaluation the following named entities are supported: person, location, organization and locorg. Sequence of assigned labels can be naturally decoded (Ratinov & Roth, 2009) to restore named entity mentions in the plain text.

To compute correct label, each token is converted into feature vector which is fed into one-vs-rest multiclass linear SVM classifier (Fan et al., 2008). The following groups of basic extractors are used to fill up feature vector of the token: word-level (Zhang & Johnson, 2003), local context and global context extractors (Ratinov & Roth, 2009).

Word-level feature extractors are used to collect pieces of information, sealed in a word itself. We use the following extractors of this group: token affixes of lengths from one to four; token text, part-of-speech tag, lemma and digit normalized token form, where all digits are replaced with a special character; predicates, indicating token properties: starting from capital letter, containing characters of the same class (digits, quotation marks), being constructed only from characters of the same class (digits, digits or letters, non-letters, uppercase letters).

Local context features encode information from nearby area of a certain token. We selected the following features of this group: token position in sentence (being first token; not being first token; not being last token); BILOU labels assigned to up to three previous tokens of the analyzed text.

The final group of features—global context features—tries to exploit information from the whole document or from some pretty large area around the token under consideration. We utilize feature values and labels distributions of tokens, sharing the same (case ignored) form among 200 previous tokens for labels and 2,000-token window around the target token for feature values.

We should also mention, that when building feature vector for a certain token, not only its features are collected—features of up to three surrounding tokens from the same sentence are appended to the vector as well.

In addition to basic, we also use two groups of more sophisticated features, which are described below.

3.1. Dictionary features

Building dictionaries is the first place, where we employed wiki-resources power for NERC task. In our approach Wikidata is used to construct a number of named entity specific dictionaries. For each of them we generate a separate feature, indicating whether it contains token under consideration or not.

To build a set of dictionaries we treat Wikidata as a property graph and select a number of seed items, characterizing current named entity type. For example, for location we select 'geographical object' (Q618123), 'historic site' (Q1081138), 'fictional location' (Q3895768); for organization—'organization' (Q43229); for person—'human' (Q5), 'fictional character' (Q95074). Then we traverse Wikidata graph along 'subclass of' edges in reverse direction (from more general to more specific class). The final step is performed along 'instance of' edge (in the reverse direction, as well), delivering a number of items, representing named entity instances of the required type (Fig. 1).

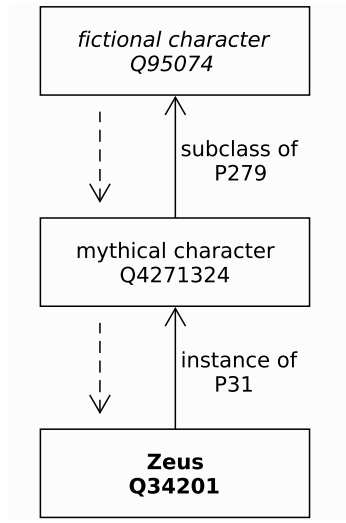

Fig. 1. Direction of Wikidata graph traversal

Labels and aliases of these instances are collected separately. The final step is to encode each entry with BILOU (actually, O is not used) and group similarly marked tokens together. Thus for each named entity type we build eight dictionaries: B, I, L and U for aliases and labels.

Additionally, we collect dictionaries of human names, surnames and nicknames. Names are gathered from vertices, linked with 'female given name' (Q11879590), 'male given name' (Q12308941), 'unisex name' (Q3409032) via 'instance of' edge; surnames—from vertices, linked with 'family name' (Q101352), 'surname' (Q4116295), 'cognomen' (Q777342), 'nomen' (Q15238609). Nicknames are collected as 'nickname' (P1449) value of vertices, denoting humans. Further steps for generating dictionaries are similar to what has already been outlined above.

3.2. Word2vec features

Another facility to utilize wiki-resources power is to build a language model. We extract plain texts from Wikipedia articles (excluding the shortest ones as they tend

to be noisy), lemmatize them and compute word embeddings with word2vec implementation (skip-gram model; hierarchical softmax enabled; window size of 10; vector dimensionality of 100). Computed vector elements are added as separate features.

3.3. Training data

When building named entity recognition model we used two separate training sets. For the first arrangement we utilized data, provided by FactRuEval organizers. As the training set was rather small (122 documents), we decided to inject more «knowledge» to the model via adding word2vec features, computed over Wikipedia.

For the second arrangement we chose to exploit only the power of Wikipedia and Wikidata mixture—FactRuEval data was abandoned. Our approach is significantly navigated by (Nothman et. al, 2013), but in contrast it requires less manual markup. Below we provide some details.

The first step here is to derive Wikipedia articles classification into named entity types. This is performed in the same way, as has been described for generating dictionaries: Wikidata graph is traversed from a number of seed vertices, denoting a single named entity type; this type is then assigned to Wikipedia articles, corresponding to each of the collected target vertices.

Further steps repeat the approach, described in (Nothman et. al, 2013): extracting text with links, mapping links to named entities, enriching text with additional mentions, selecting sentences for the train corpus.

We should point out, that our approach to constructing Wikipedia articles classification is less time-consuming, comparing to (Nothman et. al, 2013): we derive required information directly from Wikidata graph traversal instead of manually collecting initial seed set of marked up Wikipedia articles, further used to train a classifier. However, there is some disadvantage: our named entity recognizer fails to tell location from locorg, which is expected in FactRuEval-2016, due to lack of such distinction in the training data.

4. Evaluation results

In this section we present evaluation results for two NERC arrangements: FactRuEval- and Wiki-based.

4.1. FactRuEval-based arrangement

We exploit training corpus, provided by FactRuEval-2016 organizers, to build named entity recognizer, capable of distinguishing entities of the four required types: person, organization, location, locorg. To evaluate aptitude of different feature groups we examine several combinations (Table 1). Contribution of word2vec and dictionary features, used separately, looks significant; still their mixture delivers some final bits to the result.

Table 1. Feature group combinations in FactRuEval-based arrangement

feature set	precision	recall	f1
basic features	0.7357	0.6186	0.6720
basic + dictionary features	0.8098	0.6988	0.7502
basic + word2vec features	0.8093	0.7241	0.7643
basic + dictionary + word2vec features	**0.8257**	**0.7408**	**0.7810**

Table 2 contains detailed evaluation results for all-features arrangement. Our method is good for such well-defined named entity type as person, while it also provides reasonable results for other types.

Table 2. Per-type evaluation results for full feature set FactRuEval-based arrangement

type	precision	recall	f1
person	0.9340	0.8675	0.8995
location	0.7259	0.6944	0.7098
organization	0.7844	0.6548	0.7137
locorg	0.7858	0.7251	0.7542
overall	0.8257	0.7408	0.7810

4.2. Wiki-based arrangement

For training we use texts, generated from Wikidata and Wikipedia, as has been described earlier. Word2vec features, computed on Wikipedia, are not used in this arrangement. On presenting results we ignore difference between location and locorg types, as they are not distinguished in the training documents. Table 3 and Fig. 2 depict FactRuEval evaluation results for different volumes of the training corpus.

Table 3. Influence of training corpus volume on results quality

number of training documents	FactRuEval testset			FactRuEval devset		
	precision	recall	f1	precision	recall	f1
500	0.8430	0.6342	0.7238	0.8190	0.6893	0.7486
1,000	0.8476	0.6274	0.7210	0.8197	0.6826	0.7449
2,000	0.8628	0.6360	0.7323	0.8278	0.6936	0.7548
3,000	0.8655	0.6332	0.7313	0.8267	0.6880	0.7510
4,000	0.8687	0.6353	0.7339	0.8413	0.6918	0.7593
5,000	0.8763	0.6388	0.7389	0.8427	0.6898	0.7586
6,000	0.8734	0.6396	0.7384	0.8486	0.6931	0.7630
8,000	0.8817	0.6470	0.7463	0.8508	0.6927	0.7637
10,000	**0.8819**	**0.6475**	**0.7467**	**0.8525**	**0.6942**	**0.7652**

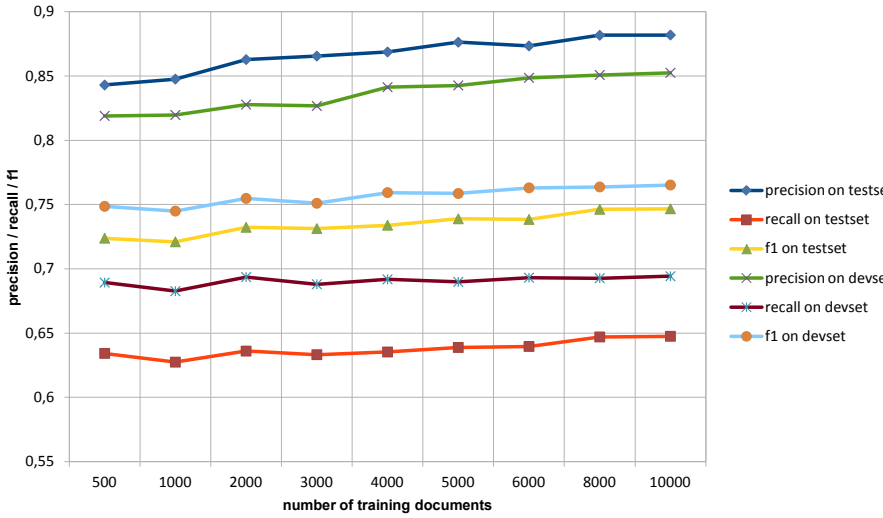

Fig. 2. Influence of training corpus volume on results quality

As one can notice, increasing volume of training corpus promotes evaluation results quality improvement, which starts to slow down at around 6,000–8,000 documents.

Table 4 reports detailed evaluation results for Wiki-based arrangement. Our method works definitely good for person named entity type; it shows promising results for other types, with the only confusing recall for organizations. We explain it with discrepancy of FactRuEval markup rules and Wikipedia linking policy: there seem to be some amount of named entities, which are not typically used as anchor texts in Wikipedia, thus making our method short on detecting them. Analysis shows that our method typically misses informally named entities ("Минфин", "Минюст", "российский газовый монополист", "парижская Третьяковка") and entities in controversial cases ("интернет", "Рунет" are considered organizations; meaning of "австралийское правительство" is not clear: governmental institution or synonym for "the authorities"; "музей Орсе" is considered organization instead of location or facility).

Table 4. Per-type evaluation results for Wiki-based arrangement (10,000 training documents) on FactRuEval-2016 testset

type	precision	recall	f1
person	0.9624	0.7976	0.8723
location	0.7993	0.7470	0.7723
organization	0.8939	0.4327	0.5831
overall	0.8819	0.6475	0.7467

Conclusion

In this paper we presented two approaches for named entity recognition and classification, both showing promising results in FactRuEval-2016 evaluation.

The first approach utilizes rich feature set: word-level, local and global context features, word2vec word embeddings and dictionary features. We showed that dictionary and word2vec features, even used separately, manage to significantly amend results quality. Combining all features together facilitated further improvement.

The second approach illustrates, how to utilize the power of Wikipedia and Wikidata mixture for automatically building a large training corpus for NERC task. Our research has shown that even a few thousand labeled documents can be used to achieve comparable results for detecting named entities of several most popular types: person, location, organization.

References

1. *Brown P. F., deSouza P. V., Mercer R. L., Pietra V. J. D., Lai J. C.* (1992), Class-based n-gram models of natural language, Computational Linguistics. Vol. 18, Issue 4, pp. 467–479.
2. *Fan R.-E., Chang K.-W., Hsieh C.-J., Wang X.-R., Lin C.-J.* (2008), LIBLINEAR: A library for large linear classification, Journal of Machine Learning Research, Vol. 9, pp. 1871–1874.
3. *Jouili S., Vansteenberghe V.* (2013), An Empirical Comparison of Graph Databases, Proceedings of the 2013 International Conference on Social Computing, Alexandria, pp. 708–715.
4. *Mikolov T., Chen K., Corrado G., Dean J.* (2013), Efficient Estimation of Word Representations in Vector Space, available at: http://arxiv.org/pdf/1301.3781.pdf
5. *Miller S., Guinness J., Zamanian A.* (2004), Name Tagging with Word Clusters and Discriminative Training, Proceedings of HLT, Boston, pp. 337–342.
6. *Milne D., Witten I. H.* (2008), Learning to link with wikipedia, Proceedings of the 17th ACM conference on Information and knowledge management (CIKM '08), Napa Valley, pp. 509–518.
7. *Nothman J., Ringland N., Radford W., Murphy T., Curran J. R.* (2013), Learning multilingual named entity recognition from Wikipedia, Artificial Intelligence, Vol. 194, pp. 151–175.
8. *Ratinov L., Roth D.* (2009), Design Challenges and Misconceptions in Named Entity Recognition, Proceedings of the Thirteenth Conference on Computational Natural Language Learning, Boulder, pp. 147–155.
9. *Ratinov L., Roth D., Downey D., Anderson M.* (2011), Local and global algorithms for disambiguation to Wikipedia, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies—Volume 1 (HLT '11), Portland, pp. 1375–1384.
10. *Rau L. F.* (1991), Extracting company names from text, Proceedings. Seventh IEEE Conference on Artificial Intelligence Applications, Miami Beach, pp. 29–32.

11. *Turdakov D., Astrakhantsev N., Nedumov Y., Sysoev A., Andrianov I., Mayorov V., Fedorenko D., Korshunov A., Kuznetsov S.* (2014), Texterra: A Framework for Text Analysis [Texterra: infrastruktura dlya analiza tekstov], Proceedings of the Institute for System Programming of RAS [Trudy ISP RAN], volume 26, Issue 1, pp. 421–438.
12. *Uchimoto K., Ma Q., Murata M. Ozaku H., Isahara H.* (2000), Named entity extraction based on a maximum entropy model and transformation rules, Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Hong Kong, pp. 326–335.
13. *Zhang T., Johnson D.* (2003), A robust risk minimization based named entity recognition system, Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, Edmonton, pp. 204–207.