First Steps in Czech Entity Linking

Michal Konkol

Department of Computer Science and Engineering
Faculty of Applied Sciences
University of West Bohemia
Univerzitní 8, 306 14 Plzeň, Czech Republic
nlp.kiv.zcu.cz
{konkol}@kiv.zcu.cz

Abstract. In this paper, we present our approach for a simplified Entity Linking task in Czech, where entity mentions found in text are linked to a list of known entities. We evaluate both known and newly proposed methods for entity names similarity on a manually annotated newspaper corpus. We show that it is possible to achieve a very high accuracy in this task, which is required in many natural language processing tasks as well as in the commercial practice.

Keywords: Entity Linking, Named Entity, Named Entity Disambiguation, Czech

1 Introduction

Named entity is an (multi-word) expressions, that identifies a single object (e.g. John Doe) from a set of semantically similar objects (e.g. persons). The most often used classes of named entities are persons, organizations, locations, or dates. Named entities from these classes often hold the key information in a document, which can be exploited in many applications. There are two natural language processing tasks associated with named entities: named entity recognition and named entity disambiguation (with entity linking subtask). The named entity recognition task identifies entities in texts and classifies them. The entity linking task links the entity mentions found in text with real entities, e.g. the entity mention 'Obama' is linked with Wikipedia page of Barack Obama, the president of the USA.

The full named entity disambiguation task is not limited only to the entity linking subtask. Other subtasks are NIL detection, and NIL clustering. NIL detection tries to detect entity mentions, that are not in the knowledge base and the NIL clustering task groups these mentions in such a way, that each group refers to only a single real entity. Another related task is entity normalization, where entities are normalized to their official name, e.g. 'USA' is normalized to 'United States of America'.

In this paper, we address the task of linking entities to a list of known entities. The known entities are not associated with any data (e.g. short bios for persons), thus it is needed to link them through string similarity. This task arise very often in the research as well as in the commercial practice. We have two main goals. First, we want to propose a method, which solves this problem. Second, we need to create a new corpus in order to evaluate the proposed approach.

2 Related Work

While the named entity recognition is well studied in Czech [1–6], the entity linking task has not been addressed yet.

The most prominent resources and systems for entity linking were introduced for the Knowledge Base Population (KBP) task of the Text Analysis Conference (TAC) [7]. The data support the full named entity disambiguation task, i.e. entity linking, NIL detection and NIL clustering. There are also data for cross-lingual entity linking [8].

Another well-known task is Web Person Search (WePS) [9]. The task is to cluster web pages returned to a person name query so that each cluster refers to a different person.

3 Corpus creation

The corpus is based on press releases from the Czech News Agency and a list of known entities. It is important to note, that with a list of known entities we only try to solve the variety problem (multiple surface forms for one entity) and not the ambiguity problem (one surface form for multiple entities). We have chosen to use only the person names, because they have (according to our estimates) higher frequency and variety in the news domain than organizations and locations.

Each entity was assigned to the corresponding entry in the list of known entities if such an entry exists. There are situations, in which the entity can be assigned to multiple entries or we are not able to decide certainly, e.g. for entity mention ‘Doe’, we cannot decide if it is ‘John Doe’ or ‘Jack Doe’ from the list of know entities and even if there is only the entry ‘John Doe’, we cannot be certain that the document is about ‘John Doe’ and not some other ‘Doe’. For this purpose, two types of links are defined: certain and possible. A certain link is used for entities that can be linked certainly to the list given the document, e.g. ‘Johnnie’ can be certainly linked to ‘John Doe’ only if it is obvious from the document (without external knowledge), that ‘Johnnie’ refers to ‘John Doe’.

We have annotated 77 documents with 879 entity mentions. The list of known entities contains 21648 entries. From the 879 found entity mentions, 316 are linked to a known entity and 563 are not linked. Certain link was assigned to 253 of the linked entities and possible link to 63 entity mentions. There were 213 possible links in total, what makes the average of more than 3 possible links for the 63 entity mentions.

The certainly linked entity mentions referenced 96 distinct entries in the dictionary. Each linked entity mention is a surface form of the particular known entity. There were 38 known entities referenced by more than one surface form, so the variety is approximately 39.5%. On average the referenced known entities have 1.9 surface forms. The most surface forms (9) and links (15) were found for ‘Ehud Olmert’, an Israel politician and former prime minister. There are multiple documents in the corpus dealing with Israel politics. Figures 1 and 2 show the histograms of the number of surface forms and links.

Fig. 1: Histogram of surface forms per entity.

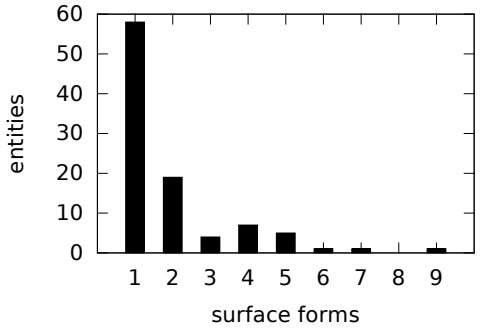
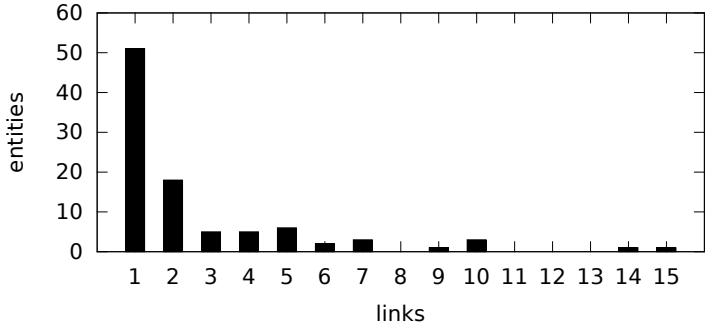


Fig. 2: Histogram of links per entity.



4 Similarity metrics

In this section, we introduce the string similarity metrics used in our experiments. Before their introduction, we need to define our mathematical notation. We compare strings a and b. The length of the string a is denoted as $|a|$. We say that A is a set of n-grams (and their counts) contained in string a. The sum of counts of all n-grams in this set is denoted as $|A|$. The union $A \cup B$ contains all n-grams of A and B, where the count of a shared n-gram is the maximum of their original counts. The intersection $A \cap B$ contains all shared n-grams and their counts are minimums of their original counts. E.g. we have $a =$ 'aaab' and $b =$ 'aabc', the sets are $A = \{('aa', 2), ('ab', 1)\}$ and $B = \{('aa', 1), ('ab', 1), ('bc', 1)\}$, $|A| = 3$ and $|B| = 3$, the union $A \cup B = \{('aa', 2), ('ab', 1), ('bc', 1)\}$ and the intersection $A \cap B = \{('aa', 1), ('ab', 1)\}$.

The *Levenshtein distance* is probably the most commonly used string distance metric. It computes the minimal edit distance using three edit operations – delete, add, and substitute. The Levenshtein metric is defined as (1), where $\mathbf{1}_{a_i \neq b_i} = 1$ if $a_i \neq b_i$ and 0 otherwise. The Levenshtein distance is converted to similarity using (2).

$$D_L(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} D_L(i-1,j)+1 \\ D_L(i,j-1)+1 \\ D_L(i-1,j-1)+\mathbf{1}_{a_i \neq b_j} \end{cases} & \text{otherwise} \end{cases} \tag{1}$$

$$S_L = 1 - \frac{D_L(|a|,|b|)}{\max\{|a|,|b|\}} \tag{2}$$

The *Levenshtein-Damerau distance* extends the set of operations in the Levenshtein distance by the transposition of adjacent characters. It is defined by (3) and converted to similarity using the same approach as for Levenshtein distance.

$$D_{LD}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} D_{LD}(i-1,j)+1 \\ D_{LD}(i,j-1)+1 \\ D_{LD}(i-1,j-1)+\mathbf{1}_{a_i \neq b_j} \\ D_{LD}(i-2,j-2)+1 \end{cases} & \text{if } i,j > 1 \text{ and } a_i = b_{j-1} \text{ and } a_{i-1} = b_j \\ \min \begin{cases} D_{LD}(i-1,j)+1 \\ D_{LD}(i,j-1)+1 \\ D_{LD}(i-1,j-1)+\mathbf{1}_{a_i \neq b_j} \end{cases} & \text{otherwise} \end{cases} \tag{3}$$

The *Jaro distance* was designed for the person names comparison and is defined by (4), where m is the number of matching characters and t is the number of transpositions. The characters are considered as matching, if they are the same and their position differs by a maximum of k characters (5). The transpositions happen if the matching characters are in different order.

$$S_J = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|a|} + \frac{m}{|b|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \tag{4}$$

$$k = \left\lfloor \frac{\max\{|a|,|b|\}}{2} - 1 \right\rfloor \tag{5}$$

The *Jaro-Winkler distance* is an improvement of the original Jaro distance. It gives higher weight to n first characters and is defined as (6). If we denote c the length of a common prefix of a and b, then $l = \max\{c, n\}$. The weight of the first characters is denoted as p, $0 \leq p \leq \frac{1}{n}$. In our experiments we use common settings $p = 0.1$ and $n = 4$. Both these metrics are named "distances", but in fact they are similarities, i.e. $0 \leq S_{J(W)} \leq 1$ and higher values are assigned to more similar strings.

$$S_{JW} = S_J + lp(1 - S_J) \tag{6}$$

The *Jaccard similarity*, *Overlap similarity*, and *Soerensen-Dice similarity*are defined by (7), (8), and (9), respectively. These similarities were not originally proposed for string similarity, but can be used for this purpose.

$$S_{Jac} = \frac{|A \cap B|}{|A \cup B|} \tag{7}$$

$$S_{O} = \frac{|A \cap B|}{\min\{|A|, |B|\}} \tag{8}$$

$$S_{SD} = \frac{2|A \cap B|}{|A| + |B|} \tag{9}$$

The *common prefix similarity* is simply a ration between the length of a common prefix c and the length of one of the strings. We can choose both minimal or maximal length of a and b (10), the is denoted in parentheses in the experiments.

$$S_{CP_{max}} = \frac{c}{\max\{|a|, |b|\}} \quad \text{or} \quad S_{CP_{min}} = \frac{c}{\min\{|a|, |b|\}} \tag{10}$$

The *longest common subsequence similarity* is the ratio of the length of the longest common subsequence lcs and the length of one of the strings. We can again use minimal or maximal length of a and b (11).

$$S_{LCS_{max}} = \frac{lcs}{\max\{|a|, |b|\}} \quad \text{or} \quad S_{LCS_{min}} = \frac{lcs}{\min\{|a|, |b|\}} \tag{11}$$

5 Proposed combination

The proposed system is based on the maximum entropy classifier. We use the implementation of this algorithm from the Brainy library [10].

We use the similarities from the previous section as features, but not directly. We firstly tokenize the entities, then we align the tokens to maximize the overall similarity. For this purpose we use a (suboptimal) greedy algorithm, which seems to be sufficient for the person names. The Hungarian algorithm [11] can be used for the optimal alignment, but has higher complexity.

A missing token (one entity has more tokens than the other) is aligned to `null` token and the similarity is set to a constant M. Furthermore, if one of the tokens is an acronym and it can represent the other token, we set the similarity to a constant R. Both R and M are parameters of the system. Using the development data, we have set these parameters to $M = 0.5$ and $R = 0.65$.

The final similarity S is the arithmetic mean of similarities between all tokens. We use the following features for all the similarity metrics:

- Similarity
- Dissimilarity $(1 - S)$
- Intervals of length 0.1 (e.g $0.3 \leq S \leq 0.4$)
- Lesser than a threshold (0.1 step, e.g $S \leq 0.4$)
- Greater or equal than a threshold (0.1 step, e.g $S \geq 0.4$)

6 Experiments

The experiments are based on *queries*, similarly to the KBP entity linking task. Each query contains a document with all entity annotations and one annotation (or entity mention) is chosen as the query. Each query is associated with the correct answer, i.e. the correct entry in the list of known entities or indication of unknown entity. There is a limited amount of positive examples (e.g. the entity mention matches the list entry), but very high number of negative examples (e.g. entity mention does not match the list entry). We have decided to use all positive examples and to select three times more negative examples, i.e. the positive examples forms $\frac{1}{4}$ of examples. We have tried to choose the hardest negative examples, where the entity mention is most similar to a wrong entry. The similarity was measured by the Levenshtein metric. This choice penalizes the Levenshtein metric when compared to other metrics as the negative examples the hardest for Levenshtein metric, but they may be easy for other metrics.

The first experiment was proposed to explore the data and to see the limits of similarity metrics. We compute a similarity s between the entity mention (query) and the list entry using each similarity metric and compare it with threshold t. If $s \geq t$, then we say that the mention matches the entry. Fig. 3 shows the relation between the chosen threshold and the accuracy. The values in parentheses are choices for the given metric (e.g. order of n-grams).

Our second experiments are done using a 10-fold cross-validation. For each fold, the data are divided in the ratio $80 : 10 : 10$ between the training, development and test data, respectively. For each similarity metric, we estimate the optimal threshold using the training data and we apply it on the test data. For the machine learning combination of similarity metrics, we use the training data to find the optimal parameters of the maximum entropy classifier, the development data to find optimal hyperparameters of the model (e.g. the optimal compensation for missing words), and we apply the best model on the test data. The results are shown in Tab. 1.

We can see, that it is possible to achieve accuracy over 90% using a simple similarity metric. The highest score using similarity metric (93.52%) was achieved with Overlap similarity using bigrams. The proposed algorithm further improves the accuracy to 97.11%. These results highly surpassed our expectations.

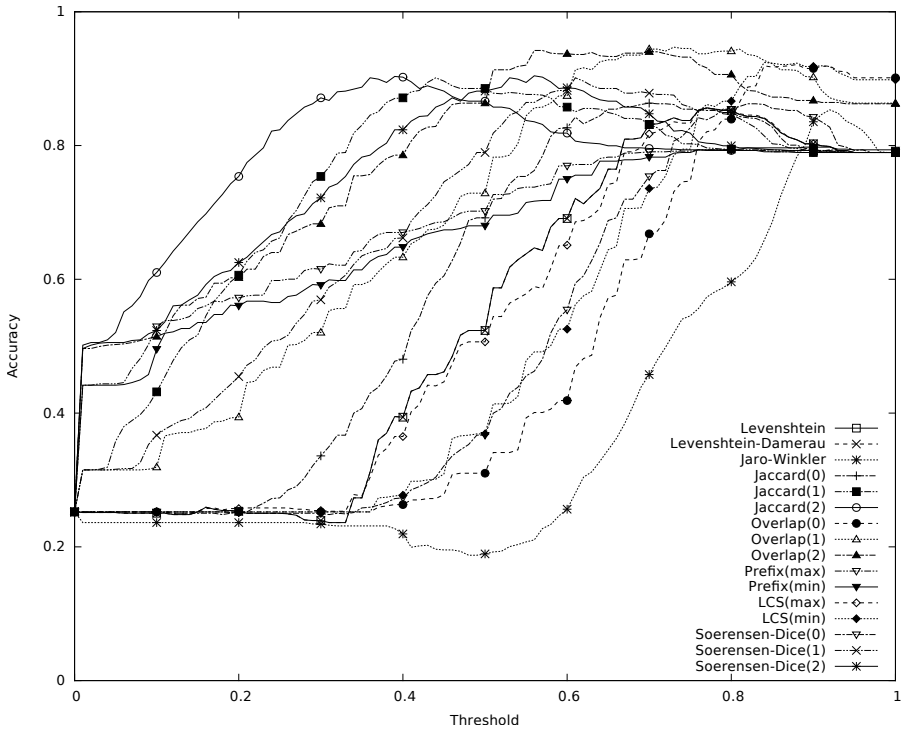
7 Conclusion and Future Work

We have manually created a Czech corpus for a simplified entity linking task and provided the necessary statistics. The data show a rather high variety (39.5%), which can be explained by the rich morphology of Czech.

We have carried out experiments with well-known similarity metrics. The best similarity metric in our experiments was Overlap similarity with accuracy 93.52%. We also propose a classifier based combination of these similarity metrics, which achieved accuracy 97.11%.

In the future, we are going to create a new corpus for the full named entity disambiguation task.

Fig. 3: Similarity metrics accuracy for various threshold settings.



References

1. Konkol, M., Brychcín, T., Konopík, M.: Latent semantics in named entity recognition. Expert Systems with Applications **42**(7) (2015) 3470 – 3479
2. Konkol, M., Konopík, M.: Maximum entropy named entity recognition for czech language. In Habernal, I., Matoušek, V., eds.: Text, Speech and Dialogue. Volume 6836 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 203–210
3. Král, P.: Features for Named Entity Recognition in Czech Language. In: KEOD. (2011) 437–441
4. Konkol, M., Konopík, M.: Crf-based czech named entity recognizer and consolidation of czech ner research. In Habernal, I., Matoušek, V., eds.: Text, Speech and Dialogue. Volume 8082 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 153–160
5. Konkol, M., Konopík, M.: Named entity recognition for highly inflectional languages: Effects of various lemmatization and stemming approaches. In Sojka, P., Hork, A., Kopeek, I., Pala, K., eds.: Text, Speech and Dialogue. Volume 8655 of Lecture Notes in Computer Science. Springer International Publishing (2014) 267–274
6. Straková, J., Straka, M., Hajič, J.: A new state-of-the-art czech named entity recognizer. In Habernal, I., Matoušek, V., eds.: Text, Speech and Dialogue: 16th International Conference, TSD 2013. Proceedings. Volume 8082 of Lecture Notes in Computer Science., Berlin / Heidelberg, Západočeská univerzita v Plzni, Springer Verlag (2013) 68–75

Table 1: Results for similarity metrics and their machine learning combination on the training, development, and test data.

Model	Accuracy		
	Training	Development	Test
Levenshtein	86.37%	84.45%	83.75%
Levenshtein-Damerau	86.37%	84.45%	83.75%
Jaro-Winkler	85.66%	84.95%	83.85%
Jaccard(0)	86.96%	85.84%	85.74%
Jaccard(1)	91.07%	88.33%	89.33%
Jaccard(2)	91.07%	89.13%	86.94%
Overlap(0)	92.71%	91.03%	90.43%
Overlap(1)	95.06%	93.82%	93.52%
Overlap(2)	94.95%	92.42%	92.22%
Prefix(max)	79.20%	79.36%	79.36%
Prefix(min)	79.55%	79.66%	79.66%
LCS(max)	86.37%	84.75%	84.65%
LCS(min)	92.71%	91.72%	91.63%
Soerensen-Dice(0)	86.96%	85.54%	85.64%
Soerensen-Dice(1)	91.07%	88.33%	89.33%
Soerensen-Dice(2)	91.19%	89.43%	87.04%
ML combination	99.79%	97.21%	97.11%

7. Simpson, H., Strassel, S., Parker, R., McNamee, P.: Wikipedia and the web of confusable entities: Experience from entity linking query creation for tac 2009 knowledge base population. In Chair), N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, European Language Resources Association (ELRA) (may 2010)
8. Ji, H., Grishman, R., Dang, H.: Overview of the TAC2011 knowledge base population track. In: TAC 2011 Proceedings Papers. (2011)
9. Artiles, J., Borthwick, A., Gonzalo, J., Sekine, S., Amig, E.: Weps-3 evaluation campaign: Overview of the web people search clustering and attribute extraction tasks. In Braschler, M., Harman, D., Pianta, E., eds.: CLEF (Notebook Papers/LABs/Workshops). (2010)
10. Konkol, M.: Brainy: A machine learning library. In Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M., eds.: Artificial Intelligence and Soft Computing. Volume 8468 of Lecture Notes in Computer Science. Springer International Publishing (2014) 490–499
11. Kuhn, H.W.: The Hungarian Method for the Assignment Problem. Naval Research Logistics Quarterly **2**(1–2) (March 1955) 83–97