On Named Entity Recognition in Targeted Twitter Streams in Polish

Jakub Piskorski
Linguistic Engineering Group
Polish Academy of Sciences
Jakub.Piskorski@ipipan.waw.pl

Maud Ehrmann
Department of Computer Science
Sapienza University of Rome
ehrmann@di.uniroma1.it

Abstract

This paper reports on some experiments aiming at tuning a rule-based NER system designed for detecting names in Polish online news to the processing of targeted Twitter streams. In particular, one explores whether the performance of the baseline NER system can be improved through the incremental application of knowledge-poor methods for name matching and guessing. We study various settings and combinations of the methods and present evaluation results on five corpora gathered from Twitter, centred around major events and known individuals.

1 Introduction

Recently, Twitter emerged as an important social medium providing most up-to-date information and comments on current events of any kind. This results in an ever-growing interest of various organizations in tools for real-time monitoring of Twitter streams to collect their business-specific information therefrom for analysis purposes. Since monitoring the entire Twitter stream appears to be unfeasible due to the high volume of published tweets, one usually monitors targeted Twitter streams, i.e., streams of tweets potentially satisfying specific information needs.

Applications for monitoring Twitter streams usually require named entity recognition (NER) capacity. However, due to the nature of Twitter messages, i.e., being short, noisy, written in an informal style, lacking punctuation and capitalization, containing misspellings, non-standard abbreviations, and non grammatically correct sentences, the application of even basic NLP tools (trained on formal texts) on tweets usually results in poor performances. In the case of well-formed texts such as online news, exploitation of contextual clues is crucial to named entity identification and classification (e.g., '*Mayor of*' in the left context of a capitalized token is a reliable pattern to classify it as city name). Such external evidence is often missing in tweets, and entity names are frequently incomplete, abbreviated or glued with other words. Furthermore, deployment of supervised ML-based techniques for NER from tweets is challenging due to the dynamic nature of Twitter.

In this paper, we report on experiments aiming at tuning a rule-based NER system, initially designed for detecting names in Polish online news, to the processing of targeted Twitter streams. In particular, we explore whether the performance of the baseline NER system can be improved through the utilization of knowledge-poor methods (based on string distance metrics) for name matching and name guessing. In comparison to English, Polish is a free-word order and highly inflective language, with particularly complex declension paradigm of proper names, which makes NER for Polish a more difficult task.

The remaining part of the paper is structured as follows. First, Section 2 provides information on related work. Next, Section 3 describes the baseline NER system and the knowledge-poor enhancements. Subsequently, Section 4 presents the evaluation results. Finally, Section 5 gives a summary and an outlook as regards future research.

2 Related Work

The problem of NER has gained lot of attention in the last two decades and a vast bulk of research on development of NER from formal texts exists (Nadeau and Sekine, 2007). Although most of the reported work focused on NER for major languages, efforts on NER for Polish have also been reported. (Piskorski, 2005) describes a rule-based NER system for Polish that covers the classical named-entity types, i.e., persons, locations, organizations, as well as numeral and temporal expres-

Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing, pages 84–93,
Sofia, Bulgaria, 8-9 August 2013.

sions. (Marcińczuk and Piasecki, 2007) and (Marcińczuk and Piasecki, 2010) report on a memory-based learning and Hidden Markov Model approach resp. to automatic extraction of information on events in the reports of Polish Stockholders, which involves NER. Also in (Lubaszewski, 2007) and (Lubaszewski, 2009) some general-purpose information extraction tools for Polish are addressed. Efforts related to creation of a dictionary of Warsaw urban proper names oriented towards NER is reported in (Savary et al., 2009; Marciniak et al., 2009). (Graliński et al., 2009) present *NERT*, another rule-based NER system for Polish which covers similar types of NEs as (Piskorski, 2005). Finally, some efforts on CRF-based NER methods for Polish are reported in (Waszczuk et al., 2010) and (Marcińczuk and Janicki, 2012).

While NER from formal texts has been well studied, relatively little work on NER for Twitter was reported. (Locke and Martin, 2009) presented a SVM-based classifier for classifying persons, locations and organizations in Twitter. (Ritter et al., 2011) described an approach to segmentation and classification of a wider range of names in tweets based on CRFs (using POS and shallow parsing features) and Labeled LDA resp. (Liu et al., 2011) proposed NER (segmentation and classification) approach for tweets, which combines KNN and CRFs paradigms. The reported precision/recall figures are significantly lower than the state-of-the-art results for NER from well-formed texts and oscillate around 50-80%. Better results were reported in case of extracting names from targeted tweets (person names from tweets on live sport events) (Choudhury and Breslin, 2011). (Nebhi, 2012) presented a rule-based NER system for detecting persons, organizations and locations which exploits an external global knowledge base on entities to disambiguate NE type. (Liu et al., 2012) proposed a factor graph-based approach to jointly conducting NER and NEN (Named Entity Normalization), which improves F-measure performance of NER and accuracy of NEN when run sequentially. An Expectation-Maximization approach to NE disambiguation problem was reported by (Davis et al., 2012). Finally, (Li et al., 2012) presented an unsupervised system for extracting (no classification) NEs in targeted Twitter streams, which exploits knowledge gathered from the web and exhibits comparable performance to the supervised approaches mentioned earlier.

Most of the above mentioned work on NER in tweets focused on English. To our best knowledge no prior work on NER in tweets in Polish has been reported, which makes our effort a pioneering contribution in this specific field. Our work also contributes to NER from targeted Twitter streams.

3 Named Entity Extraction from Targeted Tweets in Polish

The objective of this work is to explore various linguistically lightweight strategies to adapt an existing news-oriented rule-based NER system for Polish to the processing of tweets in targeted Twitter streams. Starting from the adaptation of a NER rule-based system to the processing of tweets (Section 3.1), we incrementally refine the approach with, first, the introduction of a string similarity-based name matching step (Section 3.2) and, second, the exploitation of corpus statistics and knowledge-poor method for name guessing (Section 3.3).

3.1 NER Grammar for Polish

The starting point of our explorations is an existing NER system for Polish, modeled as a cascade of finite-state grammars using the ExPRESS formalism (Piskorski, 2007). Similarly to rule-based approaches to NER for many other Indo-European languages, the grammars consist of a set of extraction patterns for person, organization and location names. The patterns exploit both internal (e.g., company designators) and external clues (e.g., titles and functions of a person, etc.) for name detection and classification; a simple extraction pattern for person names can be illustrated as follows:

```
PER :> ( ( gazetteer & [TYPE: "firstname",
                        SURFACE: #F] )
         ( gazetteer & [TYPE: "initial",
                        SURFACE: #I] ) ?
         ( surname-candidate & [SURFACE: #L] )
       ):name
  -> name: person & [NAME: #FULL-NAME]
  & #full_name := ConcWithBlanks(#F,#I,#L).
```

This rule first matches a sequence consisting of: a first name (through a gazetteer look-up), an optional initial (gazetteer look-up as well) and, finally, a sequence of characters considered as surname candidate (e.g., capitalized tokens), which was detected by a lower-level grammar[1] and is represented as a structure of type *surname-candidate*. The right-hand side of the extraction

[1]Lower-level grammar extract small-scale structures which might constitute parts of named entities.

pattern specifies the output structure of type *person* with one attribute called NAME, whose value is simply a concatenation of the values of the variables #F, #I and #L assigned to the surface forms of the matched first name, initial and surname candidate respectively.

Overall the grammar contains 15 extraction patterns for person names, 10 for location names, and 10 for organization names. It relies on a huge gazetteer of circa 294K entries, which is an extended version of the gazetteer described in (Savary and Piskorski, 2011) and includes, i.a., 39K inflected forms of both Polish and foreign first names, 86K inflected forms of surnames, 5K of organisation names (only partially inflected), 10K of inflected location names (e.g., city names, country names, rivers, etc.). No morphological analyzer for Polish is used and only a tiny fraction of the extraction patterns relies on morphological information (encoded in the gazetteer). In this original grammar, the patterns are divided into sure-fire patterns and less reliable patterns (whose precision is expected to be lower). The latter ones are patterns that rely solely on gazetteer information (simple look-up), which might have ambiguous interpretation, e.g., patterns that only match first names in text. When applied on conventional online news, the performance of this original NER grammar oscillates around 85% in terms of F-measure.

In order to process tweets, we slightly modified this grammar, mostly by simplifying it. Since mentions of entities in tweets frequently occur as single tokens (e.g., external evidence as in classical news is often missing), we did not keep the distinction between sure-fire and less-reliable patterns. Furthermore, the original NER grammar 'included' a mechanism (encoded directly in pattern specification) to lemmatize the recognized names as well as to extract various attributes such as titles (e.g., '*Pan*' (Mr.)) and position (e.g., '*Prezydent*' (president)) for persons. As we are mainly interested in the detection and classification of NEs while processing tweets, these functionalities were not needed and the grammar simply extracts names and their type. This 'reduced' NER grammar constitutes the baseline approach, and will be referred to as BASE in the remaining part of the paper. It is worth mentioning that we tested as well a version of the grammar with lower-cased lexical resources, but due to poor results (mainly due to high ambiguity of lower-case lexical entries) we did not conduct further explorations in this direction.

3.2 String distance-based Name Matching

In tweets, names are often abbreviated (e.g., '*Parl. Europ.*' and '*PE*' are abbreviations of '*Parlament Europejski*'), glued to other words (e.g., '*prezydent Komorowski*' is sometimes written as '*prezydentKomorowski*') and misspelled variants are frequent (e.g., '*Donlad Tusk*' is a frequent misspelling of '*Donald Tusk*'). The NER grammar 'as is' would fail to recognize the particular names in the aforementioned examples. Therefore, in order to improve the recall of the 'tweet grammar', we perform a second run deploying string distance metrics (in the entire targeted Twitter stream) for matching new mentions of names previously recognized by the NER grammar (see Section 3.1). Furthermore, due to the highly inflective character of Polish, we also expect to capture with string distance metrics non-nominative mentions of names (e.g., '*Rzeczpospolitej* - genitive/dative/locative form of '*Rzeczpospolita*' - the name of a Polish daily newspaper), which the NER grammar might have failed to recognize.

Inspired by the work reported in (Piskorski et al., 2009) we explored the performance of several string distance metrics. First, we tested the baseline *Levenshtein* edit distance metric given by the minimum number of character-level operations (insertion, deletion, or substitution) needed to transform one string into another (Levenshtein, 1965). Next, we used an extension thereof, namely *Smith-Waterman* (SW) metric (Smith and Waterman, 1981), which additionally allows for variable cost adjustment to the cost of a gap and variable cost of substitutions (mapping each pair of symbols from alphabet to some cost). We used a variant of this metric, where the *Smith-Waterman* score is normalized using the *Dice coefficient* (the average length of strings compared).

Subsequently, we explored variants of the *Jaro* metric (Jaro, 1989; Winkler, 1999). It considers the number and the order of the common characters between the two strings being compared. More precisely, given two strings $s = a_1 \dots a_K$ and $t = b_1 \dots b_L$, we say that a_i in s is *common* with t if there is a $b_j = a_i$ in t such that $i - R \leq j \leq i + R$, where $R = \lfloor \max(|s|, |t|)/2 \rfloor - 1$. Furthermore, let $s' = a'_1 \dots a'_{K'}$ be the characters in

s which are common with t (with preserved order of appearance in s) and let $t' = b'_1 \dots b'_{L'}$ be defined analogously. A *transposition* for s' and t' is defined as any position i such that $a'_i \neq b'_i$. Let us denote the number of transpositions for s' and t' as $T_{s',t'}$. The *Jaro* similarity is then calculated as:

$$J(s,t) = \frac{1}{3} \cdot (\frac{|s'|}{|s|} + \frac{|t'|}{|t|} + \frac{|s'| - \lfloor T_{s',t'}/2 \rfloor}{|s'|})$$

A *Winkler* variant of *Jaro* metric boosts this similarity for strings with agreeing initial characters and is calculated as:

$$JW(s,t) = J(s,t) + \delta \cdot boost_p(s,t) \cdot (1 - J(s,t))$$

where δ denotes the common prefix adjustment factor (default value is 0.1) and $boost_p(s,t) = \min(|lcp(s,t)|, p)$. Here $lcp(s,t)$ denotes the longest common prefix between s and t. Further, p stands for the upper bound of $|lcp(s,t)|^2$, i.e., up from a certain length of $lcp(s,t)$ the 'boost value' remains the same.

The *q-gram* metric (Ukkonen, 1992) is based on the intuition that two strings are similar if they share a large number of character-level q-grams. We used a variant thereof, namely *skip-gram* metric (Keskustalo et al., 2003), which exhibited better performance than any other variant of character-level q-grams based metrics. It is based on the idea that in addition to forming bigrams of adjacent characters, bigrams that skip characters are considered. *Gram classes* are defined that specify what kind of skip-grams are created, e.g. $\{0, 1\}$ class means that normal bigrams are formed, and bigrams that skip one character. In particular, we tested $\{0, 1\}$ and $\{0, 2\}$ classes. Due to the nature of Twitter we expected skip-grams to be particularly useful in our experiments.

Considering the declension paradigm of Polish we also considered the basic *CommonPrefix* metric introduced in (Piskorski et al., 2009), which is based on the longest common prefix. It is calculated as:

$$CP(s,t) = (|lcp(s,t)|)^2 / |s| \cdot |t|$$

Finally, we evaluated the performance of *longest common sub-strings* distance metric, which recursively finds and removes the longest common sub-string in the two strings compared. Let $lcs(s,t)$ denote the first longest common sub-string for s and t and let s_{-p} denote a string obtained by removing from s the first occurrence of p in s. The *LCS* metric is calculated as:

$$LCS(s,t) = \begin{cases} 0 \text{ if } |lcs(s,t)| \leq 2 \\ |lcs(s,t)| + LCS(s_{-lcs(s,t)}, t_{-lcs(s,t)}) \\ \text{otherwise} \end{cases}$$

The string distance-based name matching described in this section will be referred to as MATCH-X, with X standing for the name of the string distance metric being used.

3.3 Name Clustering

Since contextual clues for recognizing names in formal texts are often missing in tweets, we additionally developed a rudimentary name guesser to boost the recall. Let us also observe that using string distance metrics described in Section 3.2 to match all not yet captured mentions of previously recognized names might not be easy due the fact that the process of creating abbreviations in Twitter is very productive, e.g., '*Rzeczpospolita*' appears abbreviated as ' *Rzepa*', *Rzp.* or '*RP*, which are substantially different from the original name.

The main idea beyond the name guesser is based on the following assumption: given a targeted Twitter stream, if a capitalized word n-gram has a couple of 'similar' word n-grams in the same stream, most of which are not recognized as valid word forms, then such a group of n-grams word are most likely named mentions of the same entity (e.g., person, organization or location, etc.). To be more precise, the name guesser works as follows.

1. Compute $S = \{s_1, s_2, ...s_k\}$ - a set of word uni- and bigrams (cluster seeds) in the Twitter stream[3], where $frequency(s_i) \geq \phi^4$ and $character - length(s_i) \geq 3$ for all $s_i \in S$.

2. Create an initial set of singleton 'name' clusters: $C = \{C_1, C_2, \dots, C_k\}$ with $C_i = \{s_i\}$.

3. Build clusters of simmilar n-grams around the selected uni- and bigrams

[2]Here p is set to 6.

[3]The vast majority of names annotated in our test corpus are either word unigrams or bigrams (see Section 4.1.)

[4]ϕ We explored various values of this parameter, which is described in Section 4.2

using the string distance metric m: Assign each word n-gram w in the Twitter stream to at most one cluster C_j with $j \in \arg\min_{x \in \{1,2,\ldots,k\}} dist_m(s_x, w)$[5], and $dist_m(s_j, w) \leq maxDist$, where $maxDist$ is a predefined constant.

4. Iteratively merge most-simmilar clusters in C: If $\exists C_x, C_y \in C$ with $DIST(C_x, C_y) \leq DIST(C_i, C_j)$ for $i, j \in \{1, \ldots, |C|\}$[6] and $DIST(C_x, C_y) \leq maxDist$ then $C = C \setminus \{C_x, C_y\} \cup (C_x \cup C_y)$.

5. Discard 'small' clusters: $C = \{C_x \in C : |C_x| \geq 3\}$

6. Discard clusters containing high number of n-grams, whose parts are valid word forms, but not proper names: $C = \{C_x \in C : \Sigma_{w \in C_x} \frac{WordForm^*(w)}{|C_x|} \leq 0.3\}$, where $WordForm^*(w) = 1$ if all the words constituting the word n-gram w are valid word forms, but not proper names, and $WordForm^*(w) = 0$ otherwise, e.g., $WordForm^*(\text{Jan Grzyb}) = 0$ since *Grzyb* (eng. mushroom) can be interpreted as a valid word form, which is not a proper name, whereas *Jan* has only proper name interpretation.

7. Use the n-grams in the remaining clusters in C (each of them is considered to contain named mentions of the same entity) to match names in the Twitter stream through simple lexicon look-up.

For computing similarity of n-grams and merging clusters we used the *longest common substrings* (*LCS*) metric which performed on average best (in terms of F-measure) in the context of name matching (see Section 3.2 and 4). For checking whether tokens constitute valid word forms we exploited *PoliMorf* (Woliński et al., 2012), a freely available morphological dictionary of Polish, consisting of circa 6.7 million word forms, including proper names. Proper names are distinguished from other entries in the aforementioned resource.

The name guesser sketched above will be referred to as CLUSTERING. Instead of building the name clusters around n-grams, whose frequency exceeds certain threshold, we also tested building clusters around least frequent n-grams (i.e., whose frequency is ≤ 3), which will be referred to as CLUSTERING-INFRQ. The name guesser runs either independently or on top of the NER grammar described in Section 3.1 in order to detect 'new' names in the unconsumed part of the tweet collection, i.e., names recognized by the grammar are preserved. It is important to emphasize that the clustering-based name guesser only detects names without classifying them.

4 Experiments

4.1 Dataset

We have gathered tweet collections using Twitter search API[7] focusing on some major events in 2012/2013 and on famous individuals, namely: (a) Boston marathon bombings, (b) general comments on Donald Tusk, the prime minister of Poland, (c) discussion on the public comments of Antoni Macierewicz (a politician of the Law and Justice opposition party in Poland) on the Polish president crash in Smoleńsk (Russia) in 2010, (d) debate on the controversial firing of the journalist Cezary Gmyz from one of the major Polish newspapers *Rzeczpospolita* and, (e) a collection of random tweets in Polish. Each tweet collection was extracted using simple queries, e.g., *"zamach* AND *(Boston* OR *Bostonie)"* ("attack" AND "'Boston"' either in nominative of locative form) for collecting tweets on the Boston bombings. From each collection a subset of randomly chosen tweets was selected for evaluation purposes. We will refer to the latter as the *test corpus*, whereas the entire tweet collections will be referred to as the *stream corpus*.

In the stream corpus, we computed for each tweet: (a) the *text-like fraction* of its body, i.e., the fraction of the body which contains text, and (b) the *lexical validity*, i.e., the percentage of tokens in the text-like part of the body of the tweet which are valid word forms in Polish[8]. Figure 1 and 2 show the histograms for text-like fraction and lexical validity of the tweets in each collection in the stream corpus. We can observe that large portion of the tweets contains significant text-like part, which is

[5] We denote the distance between two strings x and y measured with the string distance metric m as $dist_m(x, y)$

[6] $DIST(C_x, C_y) = \Sigma_{s \in C_x} \Sigma_{t \in C_y} \frac{dist_m(s,t)}{|C_x| \cdot |C_y|}$ (average distance between strings in the two clusters)

[7] https://dev.twitter.com

[8] For computing lexical validity we used *PoliMorf* (Woliński et al., 2012), already mentioned in the previous section.

also lexically valid. Interestingly, the random collection exhibits lower lexical validity, which is due to more colloquial language used in the tweets in this collection.

Figure 1: Text-like fraction of the tweets in each collection.

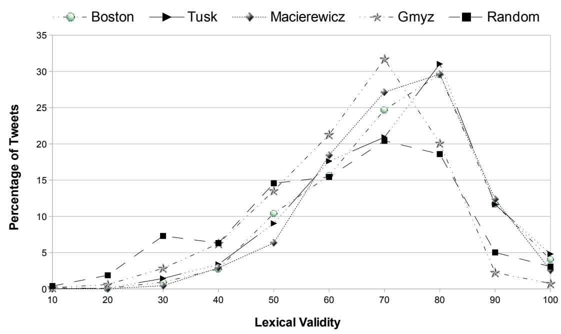

Figure 2: Lexical validity of the tweets in each collection.

We built the *test corpus* by randomly selecting tweets whose text-like fraction of the body was $\geq 80\%$, additionally checking the language and removing duplicates. These tweets were afterwards manually annotated with person, location and organization names, according to the following guidelines: consideration of unigram entities, non-inclusion of titles, functions and alike, non-inclusion of spurious punctuation marks and exclusion of names starting with '@', since their recognition as names is trivial.

The test corpus statistics are provided in Table 1. We provide in brackets the number of tweets in the corresponding tweet collections in the entire stream corpus. In this *test corpus*, 86,7% of the annotated names are word unigrams, whereas bigrams constitute 12,7% of the annotated names and 3- and 4-grams account only for a tiny fraction (0,6%); this is in line with the characteristics of the Twitter language, which favours quick and simple expressions. For each collection, we computed the name diversity as the ratio between entity occurrences and unique entities, as well as the average number of entities per tweet[9]. Targeted stream corpora show a medium name diversity (except for *Boston* and *Gmyz* collections, centred on a very specific location and person name resp.) and a high rate of entity per tweet (around 2.2), in contrast with *random* corpus which shows a high name diversity (0.79) for a low average number of entity per tweets. Reported to the limited number of characters in tweets (140), the important significant number of entity per tweet in targeted streams accounts, on the one hand, for the usefulness of working on targeted streams and, on the other, for the importance of NER in tweets.

Corpus	#tweets	name diversity	#names per tweet	#PER	#LOC	#ORG
Boston	198 (2953)	0.24	2.16	34	298	96
Tusk	232 (1186)	0.36	2.42	393	88	80
Macierewicz	303 (931)	0.32	2.17	494	60	104
Gmyz	310 (672)	0.24	2.09	471	18	159
Random	286 (7806)	0.79	0.36	59	19	27

Table 1: Test corpus statistics.

4.2 Evaluation

In our experiments we evaluated the performance of (i) the NER grammar (BASE), a combination thereof with (ii) different name matching strategies (MATCH) and (iii) different variants of the name guesser (CLUSTERING, CLUSTERING-INFRQ) and, finally, (iv) the combinations of all techniques. Within the MATCH configuration, we experimented all string distance metrics presented in 3.2 but since Jaro, Jaro-Winkler and Smith-Waterman metrics performed on average worse than the others, we did not consider them in further experiments. We selected the best performing metric, *LCS* [10], as the one used by the name guesser (CLUSTERING) in subsequent experiments. As a complement, we measured the performance of the name guesser alone to compare it with BASE. Furthermore, name matching and

[9] In the limit of our reference corpora, i.e. entities of type person, location and organization.

[10] Skip-grams was the other metric which exhibited similar performance

name guessing algorithms were using the tweet collections in the stream corpus (as quasi 'Twitter stream window') in order to gather knowledge for matching/guessing 'new' names in the test corpus.

We measured the performance of the different configurations in terms of Precision (P), Recall (R) and F-measure (F), according to two different schemes: *exact match*, where entity types and both boundaries should match perfectly, and *fuzzy match*, which allows for one name boundary returned by the system to be different from the reference, i.e., either too short or too long on the left or on the right, but not on both. Furthermore, since the clustering-based name guesser described in 3.3 does not classify names, for any settings with this technique we only evaluated name detection performance, i.e., no distinction between name types was made. The overall summary of the results for the entire pool of tweet collections, is presented in Table 3.

In the context of the CLUSTERING algorithm we explored various settings as regards the minimum frequency of an n-gram to be considered as cluster seed (ϕ parameter - see Section 3.3). More precisely, we tested values in the range of 1 to 30 for all corpora and system settings which included CLUSTERING, and compared the resulting P/R and F figures. An example of a curve with P/R values (exact match) of BASE-CLUSTERING algorithm applied on the 'Boston' corpus with varying values of ϕ is given in Figure 3. One can observe and hypothesize that the frequency threshold does not impact much the performance. Suchlike curves for other settings were of a similar nature. Therefore we decided to set the ϕ to 1 in all settings reported in Table 3.

4.3 Results analysis

The performance of the NER grammars is surprisingly good, both in case of exact and fuzzy match evaluation. Except for *random* corpus (which shows rather low performance with 55% precision and 39% recall), precision figures oscillate around 85-95%, whereas recall is somewhat worse (60-75%), as was to be expected. The low recall for 'Gmyz' corpus is due to the non-matching of a frequently occurring person name. Precision and recall figures for each entity type for BASE are given in Table 2. In general, recognition of organization names appears to be more difficult (lower recall), especially in the random corpus.

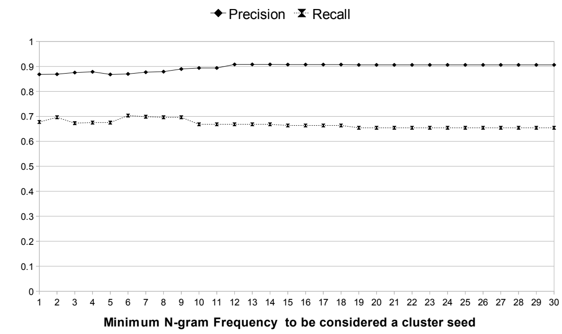

Figure 3: Precision and Recall figures for BASE-CLUSTERING applied on 'Boston' corpus, with different frequency thresholds of n-grams to be considered cluster seeds.

Corpus	PER		ORG		LOC	
	P	R	P	R	P	R
Boston	31.6	35.3	87.9	30.2	94.3	71.8
Tusk	87.6	71.2	82.4	35.0	89.9	70.5
Gmyz	85.5	32.5	82.8	15.1	88.9	44.4
Macierewicz	93.6	80.2	71.2	35.6	83.7	60.0
Random	56.7	55.9	0	0	53.3	42.1

Table 2: Precision/recall figures for person, organization and location name recognition (exact match) with BASE.

Extending BASE with MATCH yields some improvements in terms of recall (including *random* corpus), whereas precision either oscillates around the figures achieved by BASE, or deteriorates. In case of 'Gmyz' corpus, we can observe significant gain in both recall and precision through using the name matching step. With regard to the other corpora, the reason for not obtaining a significant gain could be due to two reasons: (a) the n-grams identified as similar to the names recognized by BASE are already covered by BASE with some patterns (e.g., inflected forms of many entities are stored in the gazetteer), or (b) using string distance metrics in the MATCH step might not be the best method to capture mentions of a recognized entity, as exemplified in Table 4, where the mentions of a newspaper *Rzeczpospolita* (captured by BASE) may be significantly different, e.g., in terms of the character length.

Regarding the results for CLUSTERING-INFRQ, running it alone, yielded poor results for all corpora, only in case of the'Gmyz' corpus a gain could be observed. CLUSTERING performed better than CLUSTERING-INFRQ for all corpora.

Deploying BASE with CLUSTERING on top of it results in up to 1.5-6% (exact match) and 4-

EXACT MATCH															
Method	**Boston**			**Tusk**			**Gmyz**			**Macierewicz**			**AVERAGE**		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
BASE	85.6	59.6	70.2	87.7	65.9	75.3	85.3	28.5	42.8	90.5	71.3	79.8	87.3	56.3	67.0
BASE-MATCH-LEV	80.8	62.9	70.7	87.4	66.5	75.5	90.9	63.6	**74.8**	90.2	72.3	80.3	87.3	66.3	75.3
BASE-MATCH-SW	70.9	62.1	66.3	76.6	67.5	71.8	78.0	59.1	68.0	89.4	73.1	80.4	78.7	65.5	71.6
BASE-MATCH-J	67.7	62.1	64.8	79.3	68.1	73.3	60.9	48.3	53.9	60.0	73.3	65.9	67.0	63.0	64.5
BASE-MATCH-JW	63.2	62.1	62.7	75.5	68.3	71.7	48.2	48.9	48.6	58.0	74.0	65.0	61.2	63.3	62.0
BASE-MATCH-SKIP(0,1)	80.9	62.1	70.3	87.6	66.5	75.6	91.3	63.0	74.5	90.3	72.2	80.2	87.5	66.0	75.2
BASE-MATCH-SKIP(0,2)	80.9	62.1	70.3	87.7	66.3	75.5	**91.5**	**63.0**	74.6	**90.6**	72.2	80.4	**87.7**	65.9	75.2
BASE-MATCH-CP	80.2	59.6	68.4	87.7	66.0	75.3	83.5	58.6	68.9	90.2	71.4	79.7	85.4	63.9	73.1
BASE-MATCH-LCS	80.7	63.6	71.1	86.8	67.0	75.7	82.3	59.0	68.7	90.2	72.9	80.7	85	65.6	74.1
CLUSTERING	66.2	10.0	17.4	60.6	33.2	42.9	61.3	36.0	45.3	52.9	33.4	41.0	60.3	28.2	36.7
CLUSTERING-INFRQ	37.5	1.4	2.7	27.3	1.1	2.1	60.7	31.5	41.5	54.8	28.6	37.6	45.1	15.7	21.0
BASE-CUSTERING	86.8	67.8	76.1	**91.1**	72.7	80.9	80.6	61.0	69.4	86.3	74.6	80.0	86.2	69.0	76.6
BASE-CLUSTERING-INFRQ	89.7	65.0	75.3	89.4	69.3	78.1	81.2	58.5	68.0	89.9	74.2	81.3	87.6	66.8	75.7
BASE-MATCH-CLUSTERING	87.6	**75.9**	**81.4**	90.2	**73.8**	**81.2**	74.1	62.8	68.0	86.1	**76.3**	80.9	84.5	**72.2**	**77.9**
BASE-MATCH-CLUSTERING-INFRQ	**90.0**	73.4	80.8	88.6	70.4	78.5	74.3	60.3	66.6	89.6	75.8	**82.1**	85.6	70.0	77.0

FUZZY MATCH															
Method	**Boston**			**Tusk**			**Gmyz**			**Macierewicz**			**AVERAGE**		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
BASE	86.6	60.3	71.1	92.2	69.3	79.1	88.0	29.5	44.2	95.0	74.8	83.7	90.5	58.5	69.5
BASE-MATCH-LEV	81.7	63.6	71.5	92.3	70.2	79.8	93.6	65.4	77.0	94.9	76.1	84.5	90.6	68.8	78.2
BASE-MATCH-SW	73.3	64.3	68.5	80.8	71.3	75.8	91.4	67.6	77.7	94.2	77.1	84.8	84.9	70.1	76.7
BASE-MATCH-J	70.5	64.7	67.5	85.5	73.4	79.0	86.2	68.4	76.2	63.4	77.5	69.8	76.4	71.0	73.1
BASE-MATCH-JW	65.8	64.7	65.3	81.9	74.0	77.7	68.2	69.1	68.7	61.4	78.4	68.9	69.3	71.6	70.2
BASE-MATCH-SKIP(0,1)	81.8	62.9	71.1	92.3	70.1	79.6	94.0	64.8	76.7	95.1	76.0	84.5	90.8	68.5	78.0
BASE-MATCH-SKIP(0,2)	81.8	62.9	71.1	92.2	69.7	79.4	94.2	64.8	76.8	95.0	75.7	84.3	90.8	68.3	77.9
BASE-MATCH-CP	81.1	60.3	69.2	92.2	69.3	79.1	93.8	65.9	77.4	95.0	75.2	84.0	90.5	67.7	77.4
BASE-MATCH-LCS	81.6	64.3	71.9	92.4	71.3	80.5	93.1	66.7	77.7	94.9	76.7	84.9	90.5	69.8	78.8
CLUSTERING	83.1	12.6	21.9	96.4	52.8	68.2	89.2	52.3	66.0	87.7	55.5	68.0	89.1	43.3	56.0
CLUSTERING-INFRQ	87.5	3.3	6.3	68.2	2.7	5.1	91.1	47.2	62.2	94.2	49.1	64.5	85.3	25.6	34.5
BASE-CLUSTERING	93.1	72.7	81.6	**96.9**	77.4	86.0	94.5	71.4	81.4	91.7	79.3	85.1	94.1	75.2	83.5
BASE-CLUSTERING-INFRQ	**95.5**	69.2	80.2	95.9	74.3	83.7	**96.4**	69.4	80.7	**96.9**	79.9	87.6	**96.2**	73.2	83.1
BASE-MATCH-CLUSTERING	93.3	**80.8**	**86.6**	96.5	**79.0**	**86.9**	92.9	**78.7**	**85.2**	91.8	81.3	86.2	93.6	**80.0**	**86.2**
BASE-MATCH-CLUSTERING-INFRQ	95.1	77.6	85.5	96.0	76.3	85.0	94.5	76.7	84.7	96.6	**81.8**	**88.6**	95.6	78.1	86.0

Table 3: Precision, Recall and F-measure figures for exact (top) and fuzzy match (bottom). The best results are highlighted in bold.

CEZARY GMYZ zwolniony z "**Rzeczpospolitej**". To efekt spotkania z Zarządem i Radą Nadzorczą wydawcy dziennika http://t.co/QspE3edh
@agawaa ...usiłujesz czepić sie szczegółu, gdy istota sprawy jest taka: **Rzepa**/Gmyz pitolili bez sensu.
Konflikt w **Rzepie**? Ta cała sytuacja na to wskazuje. Gmyz się nie wycofuje, a **Rzepa** jak najbardziej.
@volanowski Nowa linia: Gmyz wyrzucony z **Rzepy** czyli PO we wszystkich sprawach smoleńskich jest cacy i super. Ludzie na to nie pójda.
@TomaszSkory Być może "**Rz**" i Gmyz płacą teraz właśnie za "skróty myślowe" swoich informatorów. Dlaczego RMF nie płaci za "skróty" swoich?
Gmyz wyleciał z **RP**, a Ziemkiewicz stracił Subotnik? Nie lepiej było nieco zejść z 3.50 zł, czy chodzi o coś zupełnie innego?
Gmyz wyrzucony z "**Rzeczpospolitej**". "Dzisiaj zwolniono mnie dyscyplinarnie": Cezary Gmyz stracił pracę w "**Rzeczp**... http://t.co/ObZIxXML

Table 4: Examples of various ways of referring to a newspaper *Rzeczpospolita* in tweets.

10% (fuzzy match) gain in F-measure compared to BASE (mainly thanks to gain in recall), except 'Gmyz' corpus, where the gain is higher. The average gain over the four targeted corpora against the best combination of BASE-MATCH in F-measure is 1.3%. We observed comparable improvement for the *random* corpus. It turned out CLUSTERING often contributes to the recognition of names glued to other words and/or character sequences.

Combining BASE with MATCH-LCS and CLUSTERING/CLUSTERING-INFRQ yields further improvements against the other settings. In particular, the gain in F-measure of BASE-MATCH-CLUSTERING against BASE, measured over the four targeted corpora, is 10.9% and 16.7% for exact and fuzzy match respectively (mainly due to gain in recall).

Considering the nature of Twitter messages the average F-measure score over the four targeted corpora for BASE-MATCH-CLUSTERING, amounting to 77.9% (exact match) and 86.2% (fuzzy match) can be seen as a fairly good result. Although the difference in some of the corresponding scores for exact and fuzzy match appear substantial, it is worth mentioning that CLUSTERING algorithm often guesses name candidates that are either preceded or followed by some characters not belonging to the name itself, which is penalized in exact-match evaluation. This problem could be alleviated through deployment of heuristics to trim such 'unwanted' characters. Another source of false positives extracted by CLUSTERING is the fact that this method might, beyond person, organization and location types, recognize any kind of NEs, which, even not very frequent, is penalized since they are not present in our reference corpus.

In general, considering the shortness of names in Twitter, the major type of errors in all settings are either added or missed entities, but more rarely overlapping problems. One of the main source of errors is due to the fact that single-token names, which are frequent in tweets, often exhibit type ambiguity. Once badly recognized, these errors are propagated over the next processing steps.

5 Conclusions and Outlook

In this paper we have reported on experiments on tuning an existing finite-state based NER grammar for processing formal texts to NER from targeted Twitter streams in Polish through combining it with knowledge-poor techniques for string distance-based name matching and corpus statistics-based name guessing. Surprisingly, the NER grammar alone applied on the four test corpora (including circa 2300 proper names) yielded P, R, and F figures for exact (fuzzy) matching proper names (including: person, organization and locations) of 87.3% (90.5%), 56.3% (58.5) and 67% (69.5%) resp., which can be considered fairly reasonable result, though some variations across tweet collections could be observed (depending on the topic and how people 'tweet' about). The integration of the presented knowledge-poor techniques for name matching/guessing resulted in P, R and F figures for exact (fuzzy) matching names of 84.5% (93.6%), 72.2% (80.0) and 77.9% (86.2%) resp. (setting with best F-measure scores), which constitutes a substantial improvement against the grammar-based approach. We can observe that satisfactory-performing NER from targeted Twitter streams in Polish can be achieved in a relatively straightforward manner.

As future work to enhance our experiments, we envisage to: (a) enlarge the pool of test corpora, (b) carry out a more thorough error analysis, (c) test a wider range of string distance metrics (Cohen et al., 2003), (d) study the applicability of the particular NER grammar rules w.r.t. their usefulness in NER in targeted Twitter streams and (e), compare our approach with an unsupervised ML-approach, e.g. as in (Li et al., 2012).

Acknowledgments

The authors gratefully acknowledge the support of the ERC Starting Grant MultiJEDI No. 259234 and the Polish National Science Centre grant N N516 481940 'Diversum'.

References

Smitashree Choudhury and John Breslin. 2011. Extracting Semantic Entities and Events from Sports Tweets. In *Proceedings of the* 1^{st} *Workshop on Making Sense of Microposts (#MSM2011)*, pages 22–32.

William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. 2003. A Comparison of String Distance Metrics for Name-matching Tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03)*, pages 73–78.

Alexandre Davis, Adriano Veloso, Altigran S. da Silva, Wagner Meira, Jr., and Alberto H. F. Laender. 2012. Named Entity Disambiguation in Streaming Data. In *Proceedings of the* 50^{th} *Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 815–824, Stroudsburg, PA, USA. Association for Computational Linguistics.

Filip Graliński, Krzysztof Jassem, Michał Marcińczuk, and Paweł Wawrzyniak. 2009. Named entity recognition in machine anonymization. In *Recent Advances in Intelligent Information Systems*, pages 247–260, Warsaw. Exit.

Mathew Jaro. 1989. Advances in record linking methodology as applied to the 1985 census of Tampa Florida. *Journal of the American Statistical Society*, 84(406):414–420.

Heikki Keskustalo, Ari Pirkola, Kari Visala, Erkka Leppänen, and Kalervo Järvelin. 2003. Non-adjacent Digrams Improve Matching of Cross-lingual Spelling Variants. In *Proceedings of SPIRE, LNCS 22857, Manaus, Brazil*, pages 252–265.

Vladimir Levenshtein. 1965. Binary Codes for Correcting Deletions, Insertions, and Reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848.

Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. 2012. TwiNER: Named Entity Recognition in Targeted Twitter Stream. In *Proceedings of the* 35^{th} *International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 721–730, New York, NY, USA. ACM.

Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing Named Entities in Tweets. In *Proceedings of the* 49^{th} *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 359–367, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiaohua Liu, Ming Zhou, Furu Wei, Zhongyang Fu, and Xiangyang Zhou. 2012. Joint Inference of Named Entity Recognition and Normalization for Tweets. In *Proceedings of the* 50^{th} *Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 526–535, Stroudsburg, PA, USA. Association for Computational Linguistics.

Brian Locke and James Martin. 2009. *Named Entity Recognition: Adapting to Microblogging.* Senior Thesis, University of Colorado.

Wiesław Lubaszewski. 2007. Information extraction tools for polish text. In *Proc. of LTC'07, Poznań, Poland*, Poznań. Wydawnictwo Poznanskie.

Wiesław Lubaszewski. 2009. *Słowniki komputerowe i automatyczna ekstrakcja informacji z tekstu*. AGH Uczelniane Wydawnictwa Naukowo-Dydaktyczne, Kraków.

Michał Marcińczuk and Maciej Janicki. 2012. Optimizing CRF-Based Model for Proper Name Recognition in Polish Texts. In A. Gelbukh, editor, *CICLing 2012, Part I*, volume 7181 of *Lecture Notes in Computer Science (LNCS)*, pages 258––269. Springer, Heidelberg.

Michał Marcińczuk and Maciej Piasecki. 2007. Pattern extraction for event recognition in the reports of polish stockholders. In *Proceedings of IMCSIT–AAIA'07, Wisła, Poland*, pages 275–284.

Michał Marcińczuk and Maciej Piasecki. 2010. Named Entity Recognition in the Domain of Polish Stock Exchange Reports. In *Proceedings of Intelligent Information Systems 2010, Siedlce, Poland*, pages 127–140.

Małgorzata Marciniak, Joanna Rabiega-Wiśniewska, Agata Savary, Marcin Woliński, and Celina Heliasz. 2009. Constructing an Electronic Dictionary of Polish Urban Proper Names. In *Recent Advances in Intelligent Information Systems*. Exit.

David Nadeau and Satoshi Sekine. 2007. A Survey of Named Entity Recognition and Classification. *Lingvisticae Investigationes*, 30(1):3–26.

Kamel Nebhi. 2012. Ontology-Based Information Extraction from Twitter. In *Proceedings of the COLING 2012 IEEASM Workshop*, Mumbai, India.

Jakub Piskorski, Karol Wieloch, and Marcin Sydow. 2009. On Knowledge-poor Methods for Person Name Matching and Lemmatization for Highly Inflectional Languages. *Information Retrieval*, 12(3):275–299.

Jakub Piskorski. 2005. Named-Entity Recognition for Polish with SProUT. In *LNCS Vol 3490: Proceedings of IMTCI 2004, Warsaw, Poland.*

Jakub Piskorski. 2007. ExPRESS – Extraction Pattern Recognition Engine and Specification Suite. In *Proceedings of FSMNLP 2007.*

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 1524–1534, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Agata Savary and Jakub Piskorski. 2011. Language Resources for Named Entity Annotation in the National Corpus of Polish. *Control and Cybernetics*, 40(2):361–391.

Agata Savary, Joanna Rabiega-Wiśniewska, and Marcin Woliński. 2009. Inflection of Polish Multi-Word Proper Names with Morfeusz and Multiflex. *LNAI*, 5070.

T. Smith and M. Waterman. 1981. Identification of Common Molecular Subsequences. *Journal of Molecular Biology*, 147:195–197.

Esko Ukkonen. 1992. Approximate String Matching with q-grams and Maximal Matches. *Theoretical Computer Science*, 92(1):191–211.

Jakub Waszczuk, Katarzyna Głowińska, Agata Savary, and Adam Przepiórkowski. 2010. Tools and Methodologies for Annotating Syntax and Named Entities in the National Corpus of Polish. In *Proceedings of the International Multiconference on Computer Science and Information Technology (IMCSIT 2010): Computational Linguistics – Applications (CLA'10)*, pages 531–539.

William Winkler. 1999. The State of Record Linkage and Current Research Problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, Washington, DC.

Marcin Woliński, Marcin Miłkowski, Maciej Ogrodniczuk, Adam Przepiórkowski, and Łukasz Szałkiewicz. 2012. PoliMorf: A (not so) new open morphological dictionary for Polish. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012*, pages 860–864.