Lexical Correction of Polish Twitter Political Data

Maciej Ogrodniczuk, Mateusz Kopeć

Institute of Computer Science
Polish Academy of Sciences

Jana Kazimierza 5
01-248 Warsaw, Poland

maciej.ogrodniczuk@ipipan.waw.pl
m.kopec@phd.ipipan.waw.pl

Abstract

Language processing architectures are often evaluated in near-to-perfect conditions with respect to processed content. The tools which perform sufficiently well on electronic press, books and other type of non-interactive content may poorly handle noisy, colloquial and multilingual textual data which make the majority of communication today. This paper aims at investigating how Polish Twitter data (in a slightly controlled 'political' flavour) differs from expectation of linguistic tools and how it could be corrected to be ready for processing by standard language processing chains available for Polish. The setting includes specialised components for spelling correction of tweets as well as hashtag and username decoding.

1 Introduction

The recent massive growth in online media and the rise of user-authored content (e.g. weblogs, Twitter, Facebook) has led to challenges of how to efficiently access and interpret this unique data. Streaming online media pose completely new challenges to linguistic processing due to short message lengths and their noisier and more colloquial character. Moreover, they form a temporal stream strongly grounded in events and context. Consequently, existing language technologies for such languages as Polish, which is by no means an under-resourced language, but still under-researched in streaming media area, fall short on accuracy and scalability.

In this paper we present a component for real-time processing of data retrieved from Twitter — one of the linguistically most demanding large-scale stream medium. We limit our investigation to 'Polish political tweets', i.e. textual data coming from Twitter accounts of actors on the Polish political scene — members of parliament, political parties and government agencies. The motivation for such limitation is practical: tweets coming from official channels tend to be less noisy than the major stream but still reflect the same types of problems which appear in general settings. We investigate lexical characteristics of such content, possibilities of error correction and recognition of unknown words, construct tweet annotation chain with topic and named entity extraction and present a sample environment for visual content aggregation which can be treated as a demonstration of a language analytic environment to be used by Social Studies and Humanities. Each of the abovementioned steps poses a challenge in its own; topic extraction, for instance, requires application of multi-word unit lemmatization techniques, difficult for inflectional languages, and named entity extraction must be followed by resolution and unification of nicknames of political entities.

Another motivation for using political content is the rising role of social media among opinion-forming channels supplementing the public discourse traditionally represented by official records, paper and electronic media. With the advent of real-time social media, they are becoming the third major channel of political discourse, so tracking propagation of ideas in the public discourse and its growing fragmentation and polarisation seemed a solid motivation for development of linguistic processing chains for social data.

Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature. Proceedings, pages 115–125, Vancouver, BC, August 4, 2017.

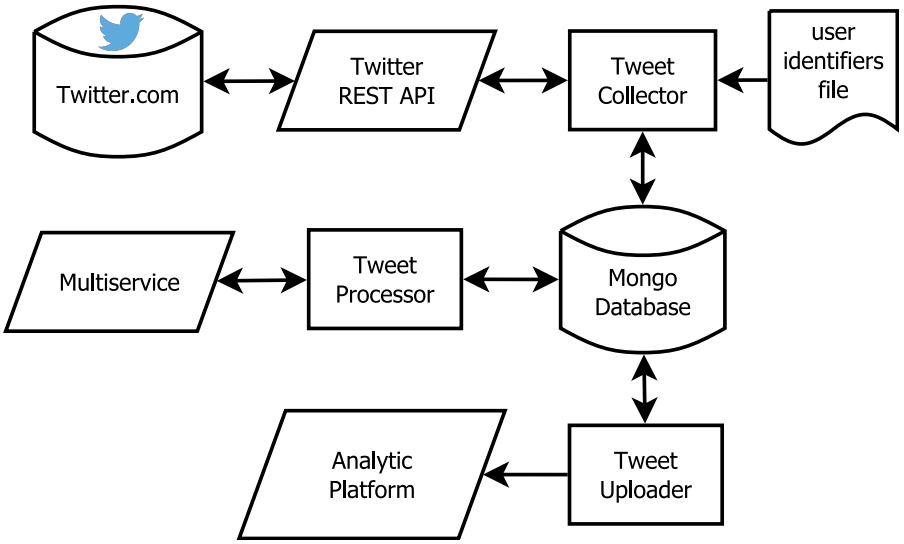

Figure 1: Architecture of the language processing chain

2 Tweet Processing Architecture

2.1 Source Data

Politics-related content was acquired from 766 accounts, the list of which was collated from several existing sources[1] and later supplemented with accounts automatically retrieved from Twitter based on the list of names of politicians and manually verified by experts to exclude fake accounts. This method is the only practical one, also due to Twitter access policy, allowing consumers to access content produced by individual accounts with a publicly available API and at the same time preventing other general access methods such as language- or topic-based filtering.

2.2 Technical Processing

The data was fetched using REST API `statuses/user_timeline` method as it provided the most complete result set (up to 3200 tweets of a given user, usually sufficient to retrieve the whole history of their publication). New activity monitoring was also set and the data was initially filtered based on detected tweet language[2]. Certain cleaning steps were also applied to the content, related to Twitter specificities (words truncated by Twitter as exceeding maximum tweet character size were removed and parts of the message indicating that a post had been retweeted were cleaned).

Twitter data was stored in the database and further processed (see Section 6) with language tools offered as Web services in a common framework called MULTISERVICE (Ogrodniczuk and Lenart, 2012)[3]. The results of the linguistic analysis were saved back to the database. The general view on the architecture of the processing pipeline is presented in Figure 1.

Twitter collection ran every hour. Since user `screen_name`[4] may change, tweets were fetched based on user identifier for users present on the account list. Date boundaries `since_id` and `max_id` were used to retrieve tweets newer than the most recent tweet from last run, and older than the oldest tweet from last request from current run. Tweets with missing identifiers or other vital information such as creation date were discarded. JSON data was stored in MongoDB. Currently the database contains over 1.7M tweets and the volume increases about 100-150K tweets/month.

Due to Twitter rate limiting which restricts free

[1]The list by Mateusz Puszczynski (no longer available online), Klub Chwila (`http://www.holdys.pl/polskitwitter/`, top 50 Twitter political accounts by `wirtualnemedia.pl` portal, data offered by ePanstwo Foundation (`http://epf.org.pl/app/webroot/api/dane/`) and manually collected list of alternative names and nicknames.

[2]Language detection was performed using OPENIMAJ library (Hare et al., 2011), see `http://www.openimaj.org`; the library reimplements the HTTPS://GITHUB.COM/SAFFSD/LANGID.PY script (Lui and Baldwin, 2012) using the model trained for 97 languages.

[3]See also `http://multiservice.nlp.ipipan.waw.pl/en/`.

[4]Detailed description of the format may be found at `https://dev.twitter.com/docs/platform-objects/tweets`.

of charge requests of each type (e.g. 300 `GET statuses/user_timeline` API calls fetching tweets of a particular user per 15 minutes), the upper bound on tweet downloading speed for the setting is 240,000 tweets per hour which is sufficient for currently available amount of Polish political Twitter content.

3 Twitter Political Language

To be able to verify a meaningful sample of Twitter political data, a 3000 tweet portion was manually inspected to perform categorization of common phenomena which could distinguish 'tweet language' from general Polish. For that purpose, 10,000 tweets were randomly selected from the stream and in the first step one tweet per each user was sampled, starting from the most recent data. Then, the sample was supplemented in a way that maintained the proportion of tweets selected to the overall number of tweets authored by a given person until the dataset reached 3000 entries. The presented method was supposed to maintain high variability in language use by providing content coming from different authors and at the same time keep a higher number of tweets authored by more active users.

The dataset, containing 39,268 words, was manually inspected to detect constructs representing pre-defined language phenomena regarded to pose a challenge for NLP tools, further referred to as 'lexical features' (LFs). The process was carried out by three project participants, each annotating 1000 entries, in one pass, with no additional verification, assuming that the task is straightforward enough to be performed without additional adjudication. For each tweet several categories of LFs were marked, shortly explained below and summarised in Table 1.

The key observation resulting from the manual analysis of political tweets is that language used in that discourse is rather well-formed and close to the quality of news articles (for example, named entities were almost always written correctly in terms of capital letters). This seems to result from rather formal character of Twitter use adopted by Polish political users, keeping this channel in line with other official means of communication. At the same time this finding nurses the hope of reasonable performance of general-purpose NLP tools on this type of data.

However, several differences between tweets and standard written communication were noticed, the most common of which was related to frequency of abbreviations. Over 26% of tweets contained at least one non-standard (out-of general dictionary) abbreviation. Another common issue were spelling errors, 10% of which were related to missing Polish diacritic marks (hardly understandable in smartphone era). Presence of emoticons (in 12% of tweets) and trimmed words (in 6% of tweets, resulting from cutting content due to tweet character limit) were another important findings.

LF category	**% of tweets with this LF**
Abbreviations	26.23%
Missing diacritical marks	10.93%
Emoticons	12.00%
Trimmed words	6.03%
Spelling errors	3.37%
Foreign language	2.93%
Other	2.77%
Case inconsistencies	1.27%
Any	49.40%

Table 1: Categorization of typical processing problems encountered by reviewing tweet content

Very interesting result of our manual analysis was that spelling errors other than missing diacritics were quite rare (present in about 3% of tweets) which also seemed specific to the observed group of official accounts expected to use 'correct' language. It also applied to case problems, which were very infrequent. This observation may lead to general conclusion that missing diacritics seem to have a different status than other type of lexical features — our users were rather careful in writing, yet not so strict about using proper diacritic signs. Foreign words occurred in less than 3% of tweets.

Summing up, NLP tools created for general Polish language should be effective for our data type, given that certain preprocessing (fixing diacritics and trimmed words, expanding abbreviations and correctly parsing emoticons) is performed. Still, extensive description of these phenomena covered by the next subsections is intended to present the complexity of the problem.

3.1 Missing Diacritical Marks

A word without diacritics may have a different meaning or no meaning at all, which increases the difficulty of text processing. Missing diacritics are subtypes of general spelling errors, but whenever adding diacritics was sufficient for getting a proper interpretation of a word (e.g. mąz → mąż), the tweet was marked with this category. Whenever both forms (with and without diacritics) were acceptable in the content, the more probable variant was selected, as in 'mblaszczak do JK powiedział chyba "prowokacja"? odczytuje z ruchu warg' (Eng. *it seems mblaszczak said 'provocation' to JK? [unclear, can be 'I am' or 'he is'] lip-reading*), when 3rd person 'odczytuje' (Eng. *he is lip-reading*) is less likely to be used in this context than 1st person 'odczytuję' (Eng. *I am lip-reading*).

Missing foreign diacritics (as in *exposé*, *Müller*) were not marked: although they are regularly applied by spellcheckers to known word forms (e.g. from *expose* to *exposé*) their English-alphabet variants are much more common since grave accents or umlauts are not easily obtainable with a standard Polish keyboard. In few cases this class also groups related problems, such as excess diacritic marks ('pisałęm' → 'pisałem') or puzzling cases foreign diacritics in Polish words ('pomarzyč' → 'pomarzyć').

3.2 Abbreviations and Trimmed Words

Almost all abbreviations were counted, including the following subcategories:

- abbreviations of named entities ('FB' → 'Facebook', 'GW' → 'Gazeta Wyborcza', 'PiS', 'PO')
- initials of people's names ('J.K.' → 'Jarosław Kaczyński', 'JVR' → 'Jan Vincent Rostowski'), also including ad-hoc abbreviations ('PDT' → 'Prime Minister Donald Tusk', 'PEK' → 'Prime Minister Ewa Kopacz')
- foreign abbreviations frequently used in Polish ('CIT', 'NATO', 'OK')
- abbreviations without the obligatory dot at the end ('nt', 'prof')
- ad-hoc abbreviations of common words, resolvable from the context ('dzienn.' → 'dziennikarz', 'dokł' → 'dokładnie')
- certain proper names, initially formed as abbreviations ('TVP', 'TVP2', 'TVN', 'CO2', but not 'ZET' in 'Radio ZET').

Some abbreviations such as Polish slang expressions, not (yet) present in the reference morphological dictionary (Saloni et al., 2015) were excluded from this group and counted as slang words ('other' group). Code or brand names ('F16', 'BMW') were also not treated as abbreviations.

Category of trimmed words resulted from users' attempts to publish longer tweets than the maximum allowed 140 characters. Trimming may occur in the middle of a word, username, hashtag or URL and is marked with triple dots, often leaving only first part of a word.

3.3 Case Problems and Spelling Errors

Category marked as case problems corresponds to entity names started with lowercase (which increases difficulty of finding named entities in the text) or unnecessary capitalization of a whole word. Lowercase letter in the beginning of the sentence was not counted as case problem.

Spelling errors category groups spelling problems other than missing diacritics into one of the following classes:

- misplaced or missing letters ('swrdecznie' → 'serdecznie', 'członkowstwo' → 'członkostwo')
- words stuck together due to missing separating spaces — excluding punctuation problems such as an extra space between a word and a comma ('gospodarkama' → 'gospodarka ma')
- words separated with an excess space ('był by' → 'byłby')
- repetitions of letters or their sequences ('okeeeeeej' → 'okej', 'Hmmmm' → ',Hmm'; not necessarily a spelling error, but not frequent enough to form a separate class).

3.4 Emoticons

Presence of emoticons may be difficult for NLP tools created for traditional written texts, such as books or news articles. Their presence requires special treatment, especially because they make a valuable source of information about the sentiment of a tweet.

3.5 Foreign Language

Most non-Polish expressions in Polish tweets were English ('dream team'), but German and Latin were also observed. Several subcategories of foreign language use can be distinguished:

- single foreign words ('community'), also those functioning as slang expressions ('sorry', 'nerd'), often inflected ('Sammyego', 'iPhone'owi')
- foreign phrases or sentences, both ad hoc interjections ('I like it') and quotations ('Ora et labora')
- titles ('Assassin's Creed Identity')
- polonized foreign words other than named entities ('retłitują' (Eng. *they retweet*, 'hendszejk' (Eng. *handshake*), 'słitfocia' (Eng. *selfie*, from *sweet photo*).

3.6 Other Phenomena

Other interesting observed phenomena included:

- neologisms ('sorkokorki', 'Kopaczinho')
- Polish ('kminię', 'Bolandzie', 'Lemingradu', 'Łomatko', 'pzdr' → 'Pozdrawiam') and foreign slang words and abbreviations ('OMG' → 'My God', 'rl' → 'real life'), sometimes noted in the dictionaries of slang
- new words, still not present in the reference morphosyntactic dictionary, but likely to be included shortly ('smartfon', 'audiobook')
- compound words, with lack of interpretation probably resulting from misconfiguration or missing prefix in the morphosyntactic dictionary ('homopropaganda', 'nadredaktor')
- less frequent forms of common words, evidently missing from the morphosyntactic dictionary ('zmolestowanego')
- non-standard transcription of common words ('nie-by-wa-łe-go')
- inflected forms of named entities, particularly adjectives ('palikotowy')
- forms intentionally distorted for stylistic reasons ('Świętokrzysko', 'szłem', 'pachły', 'wiater', 'jedenu').

4 Spelling Correction

Since our investigation showed that the most frequent lexical features in tweet content are missing diacritic marks or wrong spelling, the obvious first step of the processing was integration of an automatic spellchecker for Polish to introduce the corrections.

Spelling correction issue for Polish is a difficult task due to inflection resulting in high number of distinct word forms. PoliMorf (Woliński et al., 2012)[5], the largest morphological dictionary of Polish, contains over 44,000 lexemes corresponding to 4,000,000 word forms and 6,500,000 morphosyntactic interpretations. Without taking diacritics into consideration they are likely to be homographic. This makes such tasks as adding diacritical marks difficult in general setting. Since there is no evaluation data available, we targeted evaluation of the best available spellchecking tool for Polish — LanguageTool (Miłkowski, 2010). It is a language-independent rule-based open source proofreading software able to detect frequent context-dependent spelling mistakes, as well as grammatical, punctuation, usage, and stylistic errors. It is regarded as the most extensive resource of this type for Polish, features hundreds of thousands of downloads and is available as a standalone tool as well as a plugin for LibreOffice/OpenOffice and Firefox.

The 3000 tweet sample (see Section 3) was used as the test set for our experiment. The sample showed 740 lexical features, corresponding to misspelled words, including abbreviations and named entities ('Polasat' → 'Polsat') and words with missing diacritical marks ('zapytac' → 'zapytać'); other types of extra-lexical errors (punctuation, grammatical, usage, stylistic errors) were not taken into consideration.

The experiment showed that LanguageTool correction rules proved too extensive which resulted in introducing errors for new words ('smartfonów' → 'smart fonów'), named entities ('Baracka Obamy' → 'Baranka Obawy') and non-standard abbreviations ('pracow.' → 'placów.') in the out-of-the-box solution (referred to as version LT0 later in this section). This verification resulted in evaluating two other settings of the tool:

- running only on words which are not entirely capitalized — which corresponds to a setting

[5] See also `http://zil.ipipan.waw.pl/PoliMorf`.

	LT0	LT1	LT2	TM
Undetected errors	126	164	268	181
Detected and corrected	614	576	472	559
Wrongly corrected	695	483	178	228

Table 2: Error correction statistics for all investigated settings

	LT0	LT1	LT2	TM
Precision	0.47	0.54	0.73	0.71
Recall	0.83	0.78	0.64	0.76
F1	0.60	0.64	0.68	0.73

Table 3: Evaluation of relevance of all investigated settings

where all errors except for those in words regarded as abbreviations are corrected (setting LT1)

- running only on words which are not starting with a capital letter — which corresponds to a setting where all errors except for those in words regarded as named entities are corrected (LT2).

Taking into account the greedy behaviour or LanguageTool, another version of the spell-checking solution (later referred to as TM) was created based on the assumption that since the majority of errors are diacritic-related, fixing only this problem could solve many issues without introducing new ones likely to be caused by extensive spelling correction. TM solution implements a simple algorithm using morphosyntactic dictionary PoliMorf to extract all possible strings which by addition of some number of diacritics may represent a valid word (present in the dictionary). This gives a mapping from strings to possibilities for diacritic insertion, which produces a valid word. We also apply a special rule of not adding any diacritics if the string without them is already valid. When a string in our mapping occurs in text, we have two options: leave it unchanged, if there is such option in the mapping, or replace it with some entry from the mapping. To have an efficient way to select valid replacement (or no replacement) we use a unigram frequency count extracted from a 300-million token balanced subcorpus of the National Corpus of Polish (Przepiórkowski et al., 2012). The option which produces the most frequent word in our reference corpus is selected as valid diacritisation variant.

The algorithm works in two different modes depending on presence of diacritic signs in a tweet being corrected. If the tweet does not have any diacritics, we allow to add them if they make valid words (in this way the word 'mowie' may be corrected to 'mówię', even that it is a valid dative form of a noun 'mowa'). Otherwise, we only try to add diacritics to strings, which are not valid words.

Table 2 presents statistics of errors undetected, detected and corrected in the test data by all tools being investigated and Table 3 compares their performance. While TM solution featured more wrong corrections as compared to LT2, at the same time it detected more errors which resulted in better overall F1 score. The possible improvements of the solution might consider using context larger than unigrams or implementing a more sophisticated approach to decide whether tweet author is likely to omit diacritics or not.

5 User Name and Hashtag Normalization

Two interesting Twitter language phenomena are *hashtags* and *user mentions*. Hashtags are sequences of non-whitespace characters (making a keyword or a multi-word 'phraseword') preceded by the 'hash' (#) character, most frequently used to categorize Twitter messages by topic in the general stream of conversations (also those taking place outside writer's immediate connections). Usually composed of natural words and thus able to syntactically interact with other parts of the message, hashtags can also contribute to textual content, making them legitimate subject of natural language processing tasks. But even in case of simple keywords, their decomposition can help categorization task as in *był czechosłowacki chłopiec z plakatu to może być polska #premierzkartki* (*there was once a czechoslovakian boy from the poster so now we can have polish #primeministerfromthepieceofpaper*) where decoding reference to the Polish Prime Minister would be impossible

without hashtag segmentation.

The second useful referring feature are user mentions: by writing user account name starting with 'at' (@) sign the author can 'link' to a particular Twitter user who gets notified about that in his/her timeline which stimulates interaction. Similarly to hashtags, user names are difficult for direct use (due to their identifying rather than naming character) while at the same time they are frequently used in content not only as reference markers, but also (mostly because of the 140 character limit) as part of communication, cf. *Minister @KosiniakKamysz podaje szczegóły propozycji z expose Ewy Kopacz.* (*Minister @KosiniakKamysz gives details of proposals from Ewa Kopacz's exposé.*)

Normalization of user names is usually a simple process; they can be easily replaced with names indicated in user profiles (although more sophisticated procedures were also put forward, see e.g. (McKelvey et al., 2017)). On the contrary, multiword hashtags are often created ad hoc and are usually not camelCase-encoded so different segmentation methods should be used to process them.

Several normalization procedures have been proposed for hashtag processing, the most frequent of which follow Web domain names segmentation algorithms (Berardi et al., 2011; Wang et al., 2011; Srinivasan et al., 2012, cf. e.g.) treating it as a dictionary-based task and using the frequency distribution for selecting the most probable decomposition variant. Various hashtag harmonization methods were also proposed, e.g. by (Pöschko, 2011), based on co-occurrence of hashtags in a tweet, (Costa et al., 2013), defining meta-hashtags to be used for tweet classification, or (Declerck and Lendvai, 2015) and (Bansal et al., 2015), reducing variation of hashtags to semantically link them to topics and entities.

More recently, the topic has been adopted by a wider scientific audience, resulting in organisation of a series of workshops on tweet normalization (Tweet-Norm 2013, see `http://ceur-ws.org/Vol-1086/`), NLP for Social Media (SocialNLP), started in 2013 (see `https://sites.google.com/site/socialnlp2017/` for its 2017 edition) or Noisy User-generated Text (W-NUT) started in 2015 (see `http://noisy-text.github.io`). Proceedings of these workshops present a broad spectrum of algorithms for general social content normalization, using e.g. maximum entropy models (Arshi Saloot et al., 2015), Conditional Random Fields (Akhtar et al., 2015) or word embeddings (Costa Bertaglia and Volpe Nunes, 2016) as well as their application to languages other than English, e.g. Spanish (Pablo Ruiz, 2013) or Japanese (Ikeda et al., 2016), with numerous system resulting from a shared task on Twitter Lexical Normalization at the 2015 W-NUT.

Our normalization solution uses the PoliMorf dictionary to split hashtags into two or three parts and then select the segmentation using frequency lists from the National Corpus of Polish. Table 4 presents the results of our algorithm on the set of 1048 different hashtags identified in our test data set as unrecognized Polish words.

Result type	**Count**	**%**
Proper variant selected	**682**	**65,14%**
2 segments	573	54,73%
3 segments	109	10,41%
Wrong variant selected	**342**	**32,66%**
Foreign word	143	13,66%
New word	132	12,61%
Misspelling	14	1,34%
Unrecognized form	53	5,06%
All variants wrong	**23**	**2,20%**

Table 4: Hashtag segmentation results

Error analysis shows that the most frequent problems result from overuse of foreign words in hashtags, mostly English (*travel*, *climate* etc.); some of them tend to function as loans and are now commonly used in Polish (*tweet* (!), *startup*, *hiking*, *stalking* etc.) Several 'new words' are neologisms or newest lexical acquisitions not yet present in dictionaries (*euromajdan*, *tuskolenie*, *kartodrom*, *pendolino* (here: new Polish intercity train), *monetyzacja*, *korpo*, *polisolokaty* etc.); this category also includes frequent proper names not included in lexical database of the morphological tools such as *Obama*, *Gazprom* or *Uber*. The category of unknown words includes such forms as *indyref*, *trapani* or *himym* but also designated hashtags (cf. `#MasterChefAU`).

6 The Linguistic Platform

After spelling errors have been minimized, linguistic services integrated in MULTISERVICE can be used to perform multi-layer analysis of tweet texts. First, text is segmented and part-of-speech tagged by WCRFT (Radziszewski, 2013b)[6], a disambiguating tagger for Polish. Topics (names, locations, events) are detected using MENTION DETECTOR (Kopeć, 2014)[7], integrating data from shallow parser SPEJD (Przepiórkowski and Buczyński, 2007)[8] and named entity recognizer NERF (Savary et al., 2010)[9] Sentiment analysis is performed by Sentipejd (Buczyński and Wawer, 2008)[10]

While results of segmentation and tagging are taken over directly, results of topic detection are further categorized for visualization purposes. Firstly, nested named entities are discarded and the topic phrases are multi-word lemmatized (see details in Section 6.1). Named entities matched against Polish political ontology are additionally marked. Then noun-phrase topics are discovered using dictionary created from all previously detected mentions and their counts. When a certain, arbitrarily set count is exceeded, the phrase is marked as valid emerging topic. Overly frequent mentions such as pronouns are discarded as stopwords.

Locations are processed separately, by attempting to match lemmatized variants of each geographical named entity retrieved from tweet against Geonames ontology entries (Wick, 2015). If a match is found, GPS coordinates of that location are extracted. Twitter-offered `place` field-based location recognition results are discarded due to unclear source of the field value; according to Twitter documentation, it indicates 'a place the tweet is associated with (but not necessarily originating from)'.

Finally, token-based overall sentiment of the tweet is calculated and the bias in the Internet discourse towards negative sentiment is balanced by having a 1.5 weight in favour of the positive sentiment with the following formula:

$$S = \frac{1.5t_{pos} - t_{neg}}{t}$$

where S is the sentiment value, t is the number of tokens in tweet, t_{pos} is the number of tokens in tweet having positive sentiment and t_{neg} is the number of tokens in tweet having negative sentiment.

Visualisation and mining of data delivered by the language processing chain is further performed by a service developed by TrendMiner project (see Section 6.2).

6.1 Corpus-based Lemmatization

While lemmata for single-word expressions are provided by the tagger, lemmatization of multi-word expressions in Polish (i.e. finding the base form of a MWE) is not a trivial task, usually going far beyond word-by-word lemmatization. Citing (Graliński et al., 2010) who also lists several examples of different agreement types, this results from *complex linguistic properties of compounds, including (i) heterogeneous status of separators in the definition of a MWU's component, (ii) morphological agreement between selected components, (iii) morphosyntactic noncompositionality (exocentricity, irregular agreement, defective paradigms, variability, etc.), (iv) large sizes of inflection paradigms (e.g. dozens of forms in Polish).*

The task has been attempted previously in a narrow setting e.g. by (Degórski, 2012) or (Radziszewski, 2013a) but the results were lower than expected. As we are interested only in topic expressions which occur in multiple tweets, our approach to lemmatization of MWEs was corpus-based. The idea was to collect the number of occurrences of all MWEs (in the inflected form they occurred in text) in our Twitter database, alongside with a word-by-word lemmatization and information, whether the inflected form was analysed by NLP tools as having its syntactic head in nominative case and singular number. In such case, it is likely that the inflected form of MWE is a lemmatization of that expression. With such data, we were able to find for a MWE its lemmatized form simply by taking the most frequent inflected form (with the same word-by-word lemma as our query MWE) from the corpus, assuming we looked only at compatible base-form phrases.

This procedure was evaluated by taking 1000 random MWEs, occurring at least 10 times in our

[6] See `http://nlp.pwr.wroc.pl/redmine/projects/wcrft/wiki`.

[7] See `http://zil.ipipan.waw.pl/MentionDetector`.

[8] See `http://zil.ipipan.waw.pl/Spejd`.

[9] See `http://zil.ipipan.waw.pl/Nerf`.

[10] See `http://zil.ipipan.waw.pl/Sentipejd` and the Polish sentiment dictionary `http://zil.ipipan.waw.pl/SlownikWydzwieku`.

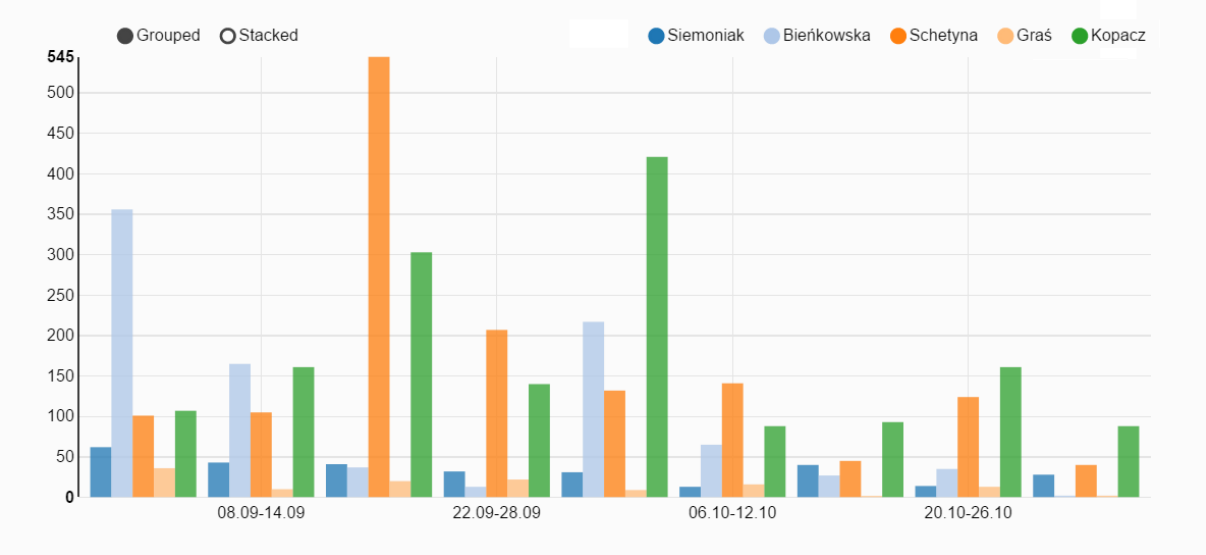

Figure 2: Numbers of mentions of entities for subperiods for five candidates to the position of Polish Prime Minister in September and October 2014

Twitter corpus, and checking validity of lemmata which were proposed for them by our algorithm. The results were encouraging, as all 1000 lemmas (!) were morphologically sound. The only issue was proper capitalization, inaccurate for 332 MWEs. For example, a common noun group may receive first capital letter, if it was most frequently used in nominative case and single number at the beginning of a sentence in our corpus. This issue, however, is less serious than presenting incorrectly inflected lemma to end users.

6.2 Political Use Case in TrendMiner Visualisation Platform

The linguistic platform was made available online in the form of an analytic portal, providing several illustrative scenarios. Figure 2 presents one of them demonstrating how Twitter reacted to the process of electing the new Prime Minister in Poland in September 2014. After it was announced on August 30 that Donald Tusk was designated as the next President of the European Council, several names for replacing him on the position of Polish Prime Minister were mentioned: Tomasz Siemoniak, Elżbieta Bieńkowska, Grzegorz Schetyna, Paweł Graś and Ewa Kopacz, who was eventually elected (on September 22). Bieńkowska, until then the Deputy Prime Minister of Poland, was leading in the first period since most political commentators regarded her as the best candidate before she was nominated as European Commissioner for Internal Market, Industry, Entrepreneurship and SMEs. Schetyna's name hit the headlines when he came back from political exile (he became a big opponent to Tusk once, so Tusk minimized his role in the party to protect his position). We can also see how Kopacz's position was constantly growing until the moment when it was decided. Then Schetyna is coming back since he was designated as the Minister of Foreign Affairs in Kopacz's government.

7 Conclusions and Future Work

The presented setting shows that even the simple methods of correcting social media content can bring improvements to language processing chains. Still, several linguistic engineering extensions of our work can be suggested. Firstly, the new lexical resources could be integrated to provide better interpretation of content, such as abbreviation dictionaries (e.g. `http://www.slownikskrotow.pl`), emoticon dictionaries (e.g. `http://krzywish.republika.pl/emotion.htm`), dictionaries of slang (e.g. `http://www.miejski.pl`) or foreign language lexicons (e.g. English aspell dictionary, see `http://aspell.net`).

More extensive interpretation of non-standard abbreviations could be integrated to handle cases where its proper interpretation is necessary for higher-level processing of content, as in 'Brawa dla PE za rez wzywającą USA do zaprzestania inwigilacji na mas skalę @panoptykon. Szkoda że w spr Snowdena PE nie zabrał głosu', ('rez' → 'rezolucję', 'mas' → 'masową', 'spr' → 'sprawie').

Two problems with this decoding is ambiguity of such abbreviations ('spr' could be equally well interpreted as 'sprawdzian' or 'sprawozdanie') and Polish inflection. Ambiguities could be resolved by context-aware corpus search of forms starting with a given prefix and proper inflected form could be generated using morphosyntactic patterns of the surrounding words.

Due to inflection of words in Polish representations of user account names and hashtags in tweet content may result in either forming grammatically incorrect phrases since hashtags and user names are usually nominative (*'Ważny tekst @ZalewskiPawel o zasadniczym dylemacie obecnej #Ukraina i roli jaka w nim przypada działaniom #Polska'*) or Twitter users inventing own methods of dealing with this problem such as adding inflection suffixes to nominative names (*'Ranking krajów najbardziej przyjaznym #senior.om.'*; it is possible only when suffixes are added and there is no alternation in the word root caused by inflection, so such addition is rather rare). As described earlier, account identifiers are replaced with user names retrieved from Twitter and hashtags decoded by replacing camelCase with spaces. However, this approach is not perfect for cases when no inflection is simulated by the user since the whole phrase must be automatically inflected (the correct version of the sentence from the first example above should read *'Ważny tekst Pawła Zalewskiego o zasadniczym dylemacie obecnej Ukrainy i roli jaka w nim przypada działaniom Polski'*). Possible solution to that problem could identify the correct case of the hashtag/user identifier in the tweet and change the case of the replacement phrase to identified case. Methods borrowed from text-to-speech synthesis systems (Graliński et al., 2007) could also be applied to produce properly inflected forms.

Acknowledgements

The work reported here was carried out within the Polish activities in the *Large-scale, Cross-lingual Trend Mining and Summarisation of Real-time Media Streams (TrendMiner)* project co-financed by the European Commission from the FP7 programme (grant agreement number 287863) and from the Polish Ministry of Science support programme (grant agreement number 3177/7.PR/2014/2).

It was also partially financed as part of the investment in the CLARIN-PL research infrastructure funded by the Polish Ministry of Science and Higher Education.

References

Md Shad Akhtar, Utpal Kumar Sikdar, and Asif Ekbal. 2015. IITP: Hybrid Approach for Text Normalization in Twitter. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 106–110, Beijing. ACL.

Mohammad Arshi Saloot, Norisma Idris, Liyana Shuib, Ram Gopal Raj, and AiTi Aw. 2015. Toward Tweets Normalization Using Maximum Entropy. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 19–27, Beijing. ACL.

Piyush Bansal, Romil Bansal, and Vasudeva Varma. 2015. Towards Deep Semantic Analysis of Hashtags. In Allan Hanbury, Gabriella Kazai, Andreas Rauber, and Norbert Fuhr, editors, *Advances in Information Retrieval — 37th European Conference on IR Research (ECIR 2015)*, volume 9022 of *Lecture Notes in Computer Science*, pages 453–464.

Giacomo Berardi, Andrea Esuli, Diego Marcheggiani, and Fabrizio Sebastiani. 2011. ISTI@TREC Microblog Track: Exploring the Use of Hashtag Segmentation and Text Quality Ranking. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of the 20th Text REtrieval Conference (TREC 2011)*, volume Special Publication 500-296. National Institute of Standards and Technology (NIST).

Aleksander Buczyński and Aleksander Wawer. 2008. Shallow parsing in sentiment analysis of product reviews. In Sandra Kübler, Jakub Piskorski, and Adam Przepiórkowski, editors, *Proceedings of the LREC 2008 Workshop on Partial Parsing: Between Chunking and Deep Parsing*, pages 14–18, Marrakech. ELRA.

Thales Felipe Costa Bertaglia and Maria das Graças Volpe Nunes. 2016. Exploring word embeddings for unsupervised textual user-generated content normalization. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 112–120, Osaka. COLING Organizing Committee.

Joana Costa, Catarina Silva, Mário Antunes, and Bernardete Ribeiro, 2013. *Defining Semantic Meta-hashtags for Twitter Classification*, pages 226–235. Springer, Berlin, Heidelberg.

Thierry Declerck and Piroska Lendvai. 2015. Processing and Normalizing Hashtags. In Galia Angelova, Kalina Bontcheva, and Ruslan Mitko, editors, *Proceedings of RANLP 2015*, pages 104–110. INCOMA Ltd.

Lukasz Degórski. 2012. Towards the Lemmatisation of Polish Nominal Syntactic Groups Using a Shallow Grammar. In Pascal Bouvry, Mieczysław A.

Kłopotek, Franck Leprevost, Małgorzata Marciniak, Agnieszka Mykowiecka, and Henryk Rybiński, editors, *Security and Intelligent Information Systems: International Joint Conference (SIIS 2011)*, volume 7053 of *Lecture Notes in Computer Science*. Springer Verlag.

Filip Graliński, Agata Savary, Monika Czerepowicka, and Filip Makowiecki. 2010. Computational Lexicography of Multi-Word Units: How Efficient Can It Be? In *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications (MWE 2010)*, pages 1–9, Beijing, China, August. ACL.

Filip Graliński, Krzysztof Jassem, Agnieszka Wagner, and Mikołaj Wypych. 2007. Linguistic Aspects of Text Normalization in a Polish Text-to-Speech System. *Systems Science*, No. 4 Vol. 32:7–15.

Jonathon S. Hare, Sina Samangooei, and David P. Dupplaw. 2011. OpenIMAJ and ImageTerrier: Java libraries and tools for scalable multimedia analysis and indexing of images. In *Proceedings of the 19th ACM international conference on Multimedia (MM 2011)*, pages 691–694, New York. ACM.

Taishi Ikeda, Hiroyuki Shindo, and Yuji Matsumoto. 2016. Japanese Text Normalization with Encoder-Decoder Model. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 129–137, Osaka. COLING Organizing Committee.

Mateusz Kopeć. 2014. Zero subject detection for Polish. In *Proceedings of the 14th Conference of the European Chapter of the ACL, volume 2: Short Papers*, pages 221–225, Gothenburg. ACL.

Marco Lui and Timothy Baldwin. 2012. Langid.Py: An Off-the-shelf Language Identification Tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Stroudsburg. ACL.

Kevin McKelvey, Peter Goutzounis, Stephen da Cruz, and Nathanael Chambers. 2017. Aligning Entity Names with Online Aliases on Twitter. In *Proceedings of the 5th International Workshop on Natural Language Processing for Social Media*, pages 25–35, Valencia. ACL.

Marcin Miłkowski. 2010. Developing an open-source, rule-based proofreading tool. *Software: Practice and Experience*, 40(7):543–566.

Maciej Ogrodniczuk and Michał Lenart. 2012. Web Service integration platform for Polish linguistic resources. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 1164–1168, Istanbul. ELRA.

Thierry Etchegoyhen Pablo Ruiz, Montse Cuadros. 2013. Lexical normalization of Spanish tweets with preprocessing rules, domain-specific edit distances, and language models. In Julio Villena Alberto Díaz, Iñaki Alegría, editor, *Proceedings of the Tweet Normalization Workshop at SEPLN 2013*, Madrid.

Jan Pöschko. 2011. Exploring Twitter Hashtags. *CoRR*, abs/1111.6553.

Adam Przepiórkowski, Mirosław Bańko, Rafał L. Górski, and Barbara Lewandowska-Tomaszczyk, editors. 2012. *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN, Warsaw.

Adam Przepiórkowski and Aleksander Buczyński. 2007. Spejd: Shallow Parsing and Disambiguation Engine. In Zygmunt Vetulani, editor, *Proceedings of the 3rd Language & Technology Conference*, pages 340–344, Poznań.

Adam Radziszewski. 2013a. Learning to lemmatise Polish noun phrases. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013). Volume 1: Long Papers*, pages 701–709. ACL.

Adam Radziszewski. 2013b. A tiered CRF tagger for Polish. In R. Bembenik, Ł. Skonieczny, H. Rybiński, M. Kryszkiewicz, and M. Niezgódka, editors, *Intelligent Tools for Building a Scientific Information Platform: Advanced Architectures and Solutions*. Springer Verlag.

Zygmunt Saloni, Marcin Woliński, Robert Wołosz, Włodzimierz Gruszczyński, and Danuta Skowrońska. 2015. Słownik gramatyczny języka polskiego. Online publication.

Agata Savary, Jakub Waszczuk, and Adam Przepiórkowski. 2010. Towards the Annotation of Named Entities in the National Corpus of Polish. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, Valletta. ELRA.

Sriram Srinivasan, Sourangshu Bhattacharya, and Rudrasis Chakraborty. 2012. Segmenting Web-domains and Hashtags Using Length Specific Models. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM 2012)*, pages 1113–1122, New York. ACM.

Kuansan Wang, Christopher Thrasher, and Bo-June Paul Hsu. 2011. Web Scale NLP: A Case Study on URL Word Breaking. In *Proceedings of the 20th International Conference on World Wide Web (WWW 2011)*, pages 357–366, New York. ACM.

Marc Wick. 2015. Geonames ontology.

Marcin Woliński, Marcin Miłkowski, Maciej Ogrodniczuk, Adam Przepiórkowski, and Łukasz Szałkiewicz. 2012. PoliMorf: a (not so) new open morphological dictionary for Polish. Istanbul. ELRA.