TweetNorm: Text Normalization on Italian Twitter Data

Daniel Weber
CIS, LMU Munich, Germany
weber.daniel@campus.lmu.de

Desislava Zhekova
CIS, LMU Munich, Germany
desi@cis.uni-muenchen.de

Abstract

This paper addresses the issue of text normalization on non-standard Italian data. We present TweetNorm[1], a system which normalizes Italian tweets in a way that the amount of microblog slang and distorted text appearance is drastically reduced and the normalized output has a much cleaner and more formal style. The paper shows that with a set of fixed language-independent rules and trained rules for language-dependent abbreviation and acronym expansion good results can be achieved for normalizing Italian Twitter messages.

1 Introduction

In general, the process of normalization of non-standard data is often a necessary preprocessing step to enable natural language processing (NLP) tools which require clean and standardized data as input to perform on their expected quality levels. Differences in the performance of NLP systems when using non-standard data instead of standard data have been early proven with the conclusion that the performance can have a decrease of up to 50% (Poibeau and Kosseim, 2001).

We focused on informal texts from a social media platform, Twitter, which holds massive amounts of user-generated data. Informal text is produced on this platform because the Twitter user's writing style changes over time and is influenced by other users and the conditions of the social service. For example, a condition on Twitter which limits the length of a tweet to 140 characters frequently leads to increased use of abbreviations of common or long words in order not to exceed the size limitation. Previous research on Twitter text normalization is mainly language-dependent and does not cover many different languages, even if generally the methods can be adapted to new languages (often with a large amount of manual effort). So far, there is no work for this task on Italian.

[1] http://www.cis.uni-muenchen.de/desi/tydiady/tweetNorm.zip

The primary goal of our work is to explore text normalization for Italian microblogs and to propose an approach that is as language-independent as possible. By using mainly general and language-independent normalization methods we show that our approach may easily be transferred to other similar, under-resourced languages.

This paper is organized as follows: After reviewing relevant related work in section 2 we describe the normalization system TweetNorm in section 3. Section 4 shows resources used by TweetNorm and provides information about the used Twitter corpus. In section 5, we evaluate the performance of TweetNorm. Section 6 concludes the paper and discusses future work.

2 Related Work

Text normalization on English data e.g. (Han and Baldwin, 2011; Li and Liu, 2012; Xue et al., 2011) is well researched compared to other languages. Especially work on microtext normalization of English and Spanish tweets was contributed as part of two workshops: W-NUT (Baldwin et al., 2015) and TWEET-NORM (Alegria et al., 2013). Most of the text normalization approaches focused on one language, mainly English. Other languages can not directly profit from these approaches as it is too time-consuming to adopt them accordingly.

The multilingual text normalization approach by Bigi (2011) which splits the normalization problem in a set of sub-problems as language-independent as possible targeted only French, English, Spanish, Vietnamese, Khmer and Chinese. Our work also follows the main idea of this approach and splits the normalization problem into various sub-problems where the language independent parts follow previous research on English Tweet normalization (Baldwin et al., 2015) based on the efficient and often used OOV (out of vocabulary) technique (Han and Baldwin, 2011) which comes naturally to mind when only processing unknown tokens.

3 Normalization via TweetNorm

This normalization process consists of three main steps. The first step identifies normalization candidates based on a standard vocabulary and tags tokens which are out of this vocabulary (OOV). The next step normalizes punctuation and OOV-tokens with regard to word contortions (spelling errors), Twitter-tags and removes OOV-tokens which are unnecessary for standardized text. All these methods are generally language-independent. To make some of these methods work in other languages too, one only needs language specific resources, such as a standardized vocabulary. Only the final step is language dependent as it normalizes OOV-tokens concerning abbreviations and acronyms.

3.1 Language-independent System Components

Multiple Character Normalization (MCN)

One part of the first normalization step targets tokens which contain repetitive characters because an author often tries to express its emotion by stretching words or punctuation marks with multiple characters. This was also recognized, described and used by Akhtar et al. (2015). All letters which appear more than two times in a row are reduced to only one occurrence. A subsequent spell checker prevents possible errors due to the reduction (*Capppuccino* → *Capuccino* → *Cappuccino*). Multiple punctuation marks are normalized similarly. There are some special cases: For example, a cascade of multiple exclamation marks and interrogation marks is always normalized to one interrogation mark. Three and more dots are always set to three dots in order not to change the meaning of the sentence.

Non-word Token Removal

The term non-word token summarizes emoticons, hyperlinks, random letter sequences, HTML markup and other special uses. There are two types of emoticon detection – a smiley gazetteer and generic smiley detection rules. In comparison to the system created by Porta and Sancho (2013) which normalizes for example *:DDDD* or *xDDDDD* to canonical emoticon forms, our system removes all detected emoticons because we only considered normalized representations for words. Hyperlinks and HTML markup allow high precision detection and also high precision adaption rules. Based on the fact that a high number of letters per token in relation to a low number of different letters typically does not represent a standard word, random letter sequences (*dfgdfgdkldkglfd*) are removed by a letter frequency algorithm, which calculates the ratio between the length of the token and single letter frequencies. It is considered to be language-independent because this algorithm does not need any training data and long words for example with a length of 16 while containing only 3 different characters are very unusal for the most Indo-European languages (comparable example: senselessnesses length:15, letters:4). The threshold of the algorithm which decides if a token is seen as random may need to be set higher for languages which do not meet these criteria because their alphabet may be smaller or more exceptions exists in this language.

In addition, a bigram language model trained on the standard vocabulary is used to identify tokens which do not represent words in this language (*iruhgcsmiegh*). The sum of letter transition probabilities in a token controls the decision process. All tokens with a low probability to be part of the language described by the model are removed by supporting methods. This is not considered as language-dependent extra work because a standard vocabulary is mandatory for each language TweetNorm is applied. Further, more than three space-separated single (capital) letters in a row are joined to detect and normalize regular words or abbreviations. Additional rules which are used to define tokens reflecting mood states which are not part of the standardized language rely on filtering and observations of the training data and may need discrete investigations for each new language. These rules include for example characteristic multiple letter tuples (*xaxaxax*, *lalala*).

Spelling Correction and String Decomposition

In order to correct words which are spelled wrong, a spelling correction was integrated, following Jin (2015) to measure similarity between two strings. The similarity of two strings is calculated with the Jaccard Index (Levandowsky and Winter, 1971) by comparing differently weighted similarity feature sets which are extracted from both strings. The vocabulary word with the highest similarity score is chosen as the correct version. Computational cost is reduced by only considering the top 150,000 frequent words with precalculated similarity feature sets for each word. In addition, only words from the tweet which have a Levenshtein distance (Levenshtein, 1966) less than three or a ratio greater

than 0.8 are considered as candidates.

To recover missing spaces between words, a string decomposing method was introduced. The method starts at the end of a long token and scans consecutive character by character for the longest match with minimal three letters. Tokens are only decomposed in case each portion of the split token represents a known word. The following example will result in two splits *Questograndeesempio* → *Questo grande esempio* while *Questo**speca**esempio* will remain without any splits because there is no proper split for *Questospeca* concerning only known words. The spelling correction was not combined with the string decomposition method in order not to accidently change the original meaning, because this raises possible splits dramatically if short tokens may be extended or potentially illformed words were corrected. However, the spelling correction was applied on hashtag splits made from capitalization patterns because the tweet author already signalized intended words with uppercase letters which reduced the number of possible splits.

Twitter Tag Normalization

Processing Twitter tags is divided in two main operations – removal and normalization. Based on the position of the tag in a tweet reliable decisions can be met. Tags which appear within the span of a tweet are usually part of the sentence structure and therefore function as a syntactic or semantic element. Starting or closing tags mostly only act as Twitter functions (user address, topic labeling) and their absence does not harm the grammar or sense of the sentence. Tags starting with an @ are resolved to personal names (*@usernameX4* → *Frank Jones*). The first level username alteration uses a dictionary of Twitter usernames mapped with its corresponding cleaned personal names. This dictionary is composed of 18 thousand name pairs seen in the training data and is extended by three top 1,000 Twitter user ranking lists. The users are ranked according to their respective number of followers, following and count of tweets.[2] The optional second level alteration rests upon live profile queries to extract, clean and save personal names. In case none of the previous layers could resolve the username, extra rules try to split the username in capitalized letter chunks (*LauraCaselli123* → *Laura Caselli*).

[2]TwitterCounter. `http://twittercounter.com`

A hashtag followed by a punctuation mark which indicates sentence boundary is always normalized and never removed because it is likely that such a hashtag might be a key element in this sentence. In the following example the search engine *Volunia* is a key element of the tweet "*why don't you switch to #volunia**?** :)*" because the hashtag cannot be removed without loosing important information. Therefore the hashtag must remain and the tweet is normalized to "*... switch to volunia?*". In this case the word *to* also signalizes that the following hashtag is embedded in the sentence and cannot be removed. These indicators can be used by taking the local word context of a hashtag into consideration. For this reason, the context of hashtags is scanned by a list of 600 Italian stop words and verbs for articles, conjunctions, prepositions and specific words which correlate syntactically or semantically with the hashtag. Based on context matches, a removal or normalization action is undertaken. The string decomposition method is also applied to hashtags in order to restore their standardized space-separated form (*#exampletopic* → *example topic*). Plenty hashtags stick to the Twitter recommendation that each new word should start with an uppercase letter. As a result if the hashtag contains a minimum of two uppercase letters it is split on capitalized letter chunks (*#FridayNight* → *friday night*). There is a possibility to feed the spelling correction with OOV chunks, but this is disabled by default because typos in hashtags rarely appeared while running the system. Almost all observed OOV chunks are named entities (organisations or names) like *Pinterest, Sgommati, Driih, Taynara* and the probability to mistakenly correct a named entity to a similiar spelled Italian word was estimated as to high with respect to the low number of necessary corrections and truely corrected words.

3.2 Language-dependent Components

Preprocessing Data

Collection and preprocessing of resources must be done for each language. Parts of the preprocessing steps are automatable but in order to achieve clean data for a new language, human work is obligatory. The preprocessing step entails the biggest effort to patch TweetNorm to a new language. The performance of TweetNorm heavily relies on the quality of the resources therefore the methods themselves do not need any patches, except special language

specific adaptions.

Abbreviation and Acronym Normalization

Each already normalized tweet was POS-tag annotated by the TreeTagger (Schmid, 1999). We used the TreeTagger since we consider the normalized tweets to be very close to standard Italian for which the TreeTagger has been originally trained. Moreover, while good POS taggers for tweets are available for English, this is not the case for Italian. All entries out of the abbreviation collection (section 4) with likely token and POS-tag context information which was extracted from training data (section 4) are initially replaced in the annotated tweet regardless whether the context matched or not. All replaced short forms in which neither the token context nor the POS-tag context matched are flagged as unsafe replacements. During a second POS-tag annotation run an algorithm decides in case of a significant increase of the context POS-tag probability compared to the first run, a POS-tag match of the full form and partial matches in previous and posterior contexts whether the unsafe replacement will be reverted or not. Entries which have no context information are seen as rare and thus they are always replaced by their unambiguous full form. Short forms which require certain conditions and patterns regarding the context like numbers or specific tokens are only replaced if all conditions are fulfilled.

4 Data Acquisition and Preparation

Standardized Vocabulary

A vocabulary which defines words that can be seen as standard is essential in this normalization approach. A good coverage of words which can be seen as standardized allow a better detection rate of normalization candidates. The vocabulary is compiled from different sources with different granularities. For further details see appendix A. All frequency lists together include more than 9,180,000 tokens. After removing smilies, punctuations, dates, numbers, links, misspelled words and other non-standard tokens plus excluding abbreviations the size of the vocabulary was reduced to 4,510,000 entries. TweetNorm supports additional user created lists (containing e.g. named entities or rare domain specific words) which can be treated as whitelists for the system to prevent unwanted token modifications. This is for example relevant if the normalization acts as preprocessing and the normalized text will later be applied to keyword sensitive applications.

Abbreviation and Acronym Collection

A collection of abbreviations and acronyms with their associated full form provides the basis for abbreviation expansion. Further information regarding the sources can be found in appendix B. The collection consists of about 400 abbreviations. Each entry was expanded by its most likely bigram contexts of tokens and bigram contexts of part-of-speech tags based on the full 9 GB POS-tagged Paisà corpus (Lyding et al., 2014). Additional acronyms were extracted from the Twitter corpus by searching for acronyms defined or mentioned within a tweet by the author. One method to find abbreviation definitions scans for keywords and punctuation which indicate mentioned abbreviations within the local context. Another context-free extraction method shrinks a tweet only to the lowercased leading letter of each token and matches sub-sequences from the original tweet:

"*This is a **s**mall **e**xample called **se**!*" → *tia**se**c*!

Twitter Corpus

The data used in this project is mainly self-procured. Periodically crawled microblogs via the Twitter REST API[3] form the biggest part of the corpus consisting of 100 thousand tweets, obtained from September to November 2015. The tweets were crawled by querying messages which contain words out of a predefined most frequent Italian word list which do not occur in any other language. All matches were post-processed with LangID (Lui and Baldwin, 2012) to assure that the messages are Italian only.

Due to limitations with Twitter's free API also mentioned by Weller et al. (2013) relating to accessibility and availability of the Twitter messages beyond a certain time frame it is hard to achieve a diversified Twitter corpus in a fast and efficient way. However, to build a corpus which is not limited to a certain time frame parts of the corpus rely on previous work done by Basile and Nissim (2013) and Basile et al. (2014). In this way, the crawled Twitter corpus was enriched with 75 thousand Italian Twitter messages from different months in 2012 and 2013 obtained by their tweet-ID which was provided by the SENTIPOLC (Basile et al., 2014) and the TWITA corpus (Basile and Nissim, 2013).

[3]Twitter. REST API. `https://dev.twitter.com`

5 Results

Unfortunately, no gold standard dataset was available for this language and task. Thus, we manually evaluated and analysed the system performance on one hundred random tweets from a set aside. Due to lack of time this set is very small and we will approach its extension as soon as possible. For each tweet, all changes done by the normalization system were manually validated. The three main categories of system applied operations cover deletion, transformation and insertion. The correctness of each operation was controlled and the resulting F_1-Score for each operation can be seen in table 1. The

Operation	Accuracy	Precision	Recall	F1
Deletion	98.42	97.31	90.95	94.02
Transformation	98.82	93.04	92.24	92.64
Insertion	99.93	97.82	100.00	98.90
Tokenization	99.37	95.30	98.61	96.69
Total				**95.56**

Table 1: Evaluation of TweetNorm operations.

transformation operation has the lowest F_1-Score with 92.64, but this operation also contains the most complex normalization methods like spelling correction, Twitter tag and abbreviation normalization. The parameters of the spelling correction are set to perform safe transformations in order to maximize precision, but there are still transformation errors. For example neologism like the portmanteau word "*twittatore*" (probably a blend of "***twitt**er*" and "***d**ittatore*") is normalized to "*dittatore*", because the morphological overlap of the involved words is too high for the parameters' sensitivity. Besides this, current errors done by the normalization of Italian abbreviations comprise incorrect expansions of unknown ambiguities like the expansion of "*San val*" to "*San valuta*" instead of "*San Valentino*". Compared to the success of the other normalization methods it still could be improved with more training data or with more specific Italian grammar knowledge and abbreviation creation rules.

After solely applying deletion operations (F_1-Score of 94.02), which includes the removal of smilies, emotionalized tokens, hyperlinks, non-semantic Twitter tags and other non-standard tokens a tweet looks much more structured and is far more readable. In addition, operations like the MCN which achieves excellent results also contribute a big part in improving the readability of a tweet. On an average this method produces 2,400 operations per 20,000 tweets out of a total of 45,000 normalization operation of all methods altogether. The manual validation indicated that the normalization of Twitter mention tags and hashtags is very robust and yields reliable output. Appendix C shows example tweets normalized by TweetNorm.

In conclusion the system performs a suitable normalization with a total F_1-Score of 95.56 on Italian tweets and the output is very similar to handmade changes.

6 Conclusion and Future Work

In this work, an approach to normalize Italian tweets according to their non-standard nature was presented which showed that is possible to achieve clean and accurate outputs with a set of language-independent rules with partial language-specific shapes relying mainly on structured resources while keeping the system itself portable and adaptable to other similar, under-resourced languages.

In order to further increase the usability and customization of TweetNorm it is planed to extend the modular design to allow easy normalization method combinations to fit individual task needs. For example turning off normalization of multiple punctuation marks for opinion mining, because sentence boundaries often indicate strong opinions. In the future this system may be used to generate training data for a statistical machine translation system (SMTS) like Moses[4] which might enhance the normalization process. The normalization task can then be seen as a machine translation problem which processes a parallel corpus of non-standard tweets and normalized tweets to extract generalized normalization rules.

References

Md Shad Akhtar, Utpal Kumar Sikdar, and Asif Ekbal. 2015. IITP: Hybrid Approach for Text Normalization in Twitter. *ACL-IJCNLP 2015*, page 106.

Iaki Alegria, Nora Aranberri, Víctor Fresno, Pablo Gamallo, Lluis Padró, Iaki San Vicente, Jordi Turmo, and Arkaitz Zubiaga. 2013. Introducción a la tarea compartida Tweet-Norm 2013: Normalización léxica de tuits en espaol. In *Workshop on Tweet Normalization at SEPLN (Tweet-Norm)*, pages 36–45.

Timothy Baldwin, Young-Bum Kim, Marie Catherine de Marneffe, Alan Ritter, Bo Han, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization

[4]Moses. `http://www.statmt.org/moses/`

and named entity recognition. *ACL-IJCNLP 2015*, page 126.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Valerio Basile and Malvina Nissim. 2013. Sentiment analysis on italian tweets. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 100–107, Atlanta.

Valerio Basile, Andrea Bolioli, Malvina Nissim, Viviana Patti, and Paolo Rosso. 2014. Overview of the evalita 2014 sentiment polarity classification task. *Proceedings of the 4th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA14). Pisa, Italy*.

Brigitte Bigi. 2011. A multilingual text normalization approach. In *5th Language & Technology Conference-The 2nd LRL WORKSHOP*, pages 1–5.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378, Portland, Oregon, USA, June. Association for Computational Linguistics.

Ning Jin. 2015. Ncsu-sas-ning: Candidate generation and feature engineering for supervised lexical normalization. *ACL-IJCNLP 2015*, page 87.

Michael Levandowsky and David Winter. 1971. Distance between sets. *Nature*, 234(5323):34–35.

VI Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

Chen Li and Yang Liu. 2012. Normalization of text messages using character-and phone-based machine translation approaches. In *INTERSPEECH*, pages 2330–2333.

Marco Lui and Timothy Baldwin. 2012. Langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, pages 25–30, Stroudsburg, PA, USA. Association for Computational Linguistics.

Verena Lyding, Egon Stemle, Claudia Borghetti, Marco Brunello, Sara Castagnoli, Felice DellOrletta, Henrik Dittmann, Alessandro Lenci, and Vito Pirrelli. 2014. The paisa corpus of italian web texts. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, pages 36–43.

Thierry Poibeau and Leila Kosseim. 2001. Proper name extraction from non-journalistic texts. In *In Computational Linguistics in the Netherlands*, pages 144–157.

Jordi Porta and José-Luis Sancho. 2013. Word Normalization in Twitter Using Finite-state Transducers. *Tweet-Norm@ SEPLN*, 1086, pages 49–53. Citeseer.

Helmut Schmid. 1999. Improvements in part-of-speech tagging with an applicationto german. In Susan Armstrong, Kenneth Church, Pierre Isabelle, Sandra Manzi, Evelyne Tzoukermann, and David Yarowsky, editors, *Natural Language Processing Using Very Large Corpora*, volume 11 of *Text, Speech and Language Processing*, pages 13–26. Kluwer Academic Publishers, Dordrecht.

Katrin Weller, Axel Bruns, Jean Burgess, Merja Mahrt, and Cornelius Puschmann. 2013. *Twitter and society*, volume 89. Peter Lang New York.

Zhenzhen Xue, Dawei Yin, and Brian D Davison. 2011. Normalizing microtext. In *Proceedings of the AAAI-11 Workshop on Analyzing Microtext*, pages 74–79.

Eros Zanchetta and Marco Baroni. 2005. Morph-it! a free corpus-based morphological resource for the italian language. *Corpus Linguistics 2005*, 1(1).

Appendices

A Vocabulary Resources

- **OpenSubtitle:** An italian frequency list build from "OpenSubtitle" corpus.[5,6]
- **Morph-it!:** A morphological resource with 31,955 Italian lemmas and 506.827 word forms (Zanchetta and Baroni, 2005).
- **itWaC:** A frequency list extracted from the "itWaC" corpus, which contains 2 billions tokens crawled from Italian websites (Baroni et al., 2009).
- **Paisà:** A frequency list based on Paisà corpus which holds 250 millions words extracted from Italian internet documents (Lyding et al., 2014).
- **ItWikiArticles:** Self-produced frequency list of all Italian Wikipedia articles up to and including september 2015.[7]

[5] OPUS. The open parallel corpus. `http://opus.lingfil.uu.se/`

[6] Invoke IT Blog. Frequency Word Lists. `https://invokeit.wordpress.com/frequency-word-lists/`

[7] Wikimedia. Italian Wikidump progress on 20151002. `https://dumps.wikimedia.org/itwiki/20151002/`

B Abbreviation Resources

- Nicla A. Gargano, Corsi ditaliani. Abbreviazioni.
  `http://homes.chass.utoronto.ca/~ngargano/corsi/corrisp/abbreviazioni.html`
- Dr. Ulrich Hondelmann, Italianita. Italian Abbreviations.
  `http://www.italianita.de/files/italienische-abkuerzungen.htm`
- PONS. Italian-German A-Z.
  `http://de.pons.com/bersetzung/italienisch-deutsch/-/A`
- Abbreviations, STANDS4 Network. Italian Abbreviations.
  `http://www.abbreviations.com/acronyms/ITALIAN`
- Michael San Filippo, About Education. Italian Abbreviations and Acronyms.
  `http://italian.about.com/od/gamespuzzles/a/aa082802a.htm`
- Foreign Broadcast Information Service. Abbreviations used in the press of Italy.
  `http://www.ut.ngb.army.mil/clp/linguists/fbis/ita.pdf`
- Andrea Sapuppo, Scuolissima. Abbreviazioni italiane.
  `http://www.scuolissima.com/2012/04/abbreviazioni-italiane.html`
- An abbreviation list created by an Italian native speaker.

C Normalization Examples

Tweets normalized by TweetNorm:
Normalization candidates in the orignal tweets and actual normalizations in the processed tweets are underlined.

- Example 1:

@marie455 Xke 6 triste :-(? tvtb :-* ,all. il n/ video con @LCuccello da nov 2014 https://youtu.be/x5PeQrRsqFo

Perch sei triste ? Ti voglio tanto bene , allegati il nostro video con Laura Cuccello da novembre 2014

- Example 2:

Quella di domaaani sar una luuuuuuunga giooornaaata!!!
Quella di domani sar una lunga giornata!

- Example 3:

Ventura, lei che maestro x i giovan8, un consiglio x Pobb?
Ventura, lei che maestro per i giovanotto, un consiglio per Pobb ?

- Example 4:

Twit della #Buonotte! :)
Twit della buonanotte!

- Example 5:

Mettersi a scaricare plugin per #photoshop alle 4 del mattino ed installarli alla prima. #mammamia!!
Mettersi a scaricare plugin per photoshop alle 4 del mattino ed installarli alla prima. Mamma mia!

- Example 6:

@heyitsflavia13 ti voglio taaaanti bene *___* sogzjlkdkdjaddfghk *attacco di dolcezza* http://t.co/SzUzdEPc
ti voglio tanto bene attacco di dolcezza