Norm It! Lexical Normalization for Italian and Its Downstream Effects for Dependency Parsing

Rob van der Goot[◇], Alan Ramponi[◇•], Tommaso Caselli[♣], Michele Cafagna[♣♠], Lorenzo De Mattei[♠]
IT University of Copenhagen[◇], University of Trento[•], University of Pisa[♠], University of Groningen[♣]
robv@itu.dk, alan.ramponi@unitn.it, {t.caselli|m.cafagna}@rug.nl,
lorenzo.dematteі@di.unipi.it

Abstract

Lexical normalization is the task of translating non-standard social media data to a standard form. Previous work has shown that this is beneficial for many downstream tasks in multiple languages. However, for Italian, there is no benchmark available for lexical normalization, despite the presence of many benchmarks for other tasks involving social media data. In this paper, we discuss the creation of a lexical normalization dataset for Italian. After two rounds of annotation, a Cohen's kappa score of 78.64 is obtained. During this process, we also analyze the inter-annotator agreement for this task, which is only rarely done on datasets for lexical normalization, and when it is reported, the analysis usually remains shallow. Furthermore, we utilize this dataset to train a lexical normalization model and show that it can be used to improve dependency parsing of social media data. All annotated data and the code to reproduce the results are available at: http://bitbucket.org/robvanderg/normit.

Keywords: Corpus (Creation, Annotation, etc.), Parsing, Grammar, Syntax, Treebank, Social Media Processing, Italian, Normalization.

1 Introduction

Social media provide a rich source of constant information, which can be used for many purposes. Italian is one of the most popular languages on the Internet, estimated to be the 9th most popular by w3techs,[1] and was the 13th most popular language on Twitter in 2018.[2] However, many online sources are much harder to process automatically, not only because many existing tools are designed with canonical texts in mind, but also because they naturally contain more linguistic variety (Eisenstein, 2013; Plank, 2016). For social media data, this performance drop was observed for multiple tasks; for POS tagging, accuracy dropped from 97% (Toutanova et al., 2003) on news data to 85% (Gimpel et al., 2011) for tweets, whereas for dependency parsing, performance dropped from 91% (Kiperwasser and Goldberg, 2016) to 62% (van der Goot and van Noord, 2018). Even when in-domain training data is available, the performance typically remains much lower compared to newswire texts, as shown by Liu et al. (2018).

One solution to this problem is to translate social media language to canonical language, a task also known as lexical normalization. An example of normalization for the sentence "joker cmq nn e' nnt di ke !" is shown in Figure 1. The example shows that a variety of phenomena is involved and must be properly annotated: (i) vowels omitted in multiple words (e.g. nn ↦ *non*); (ii) diacritics not correctly used (e' ↦ *è*); (iii) orthography based on pronunciation (ke ↦ *che*).

Previous work has shown that lexical normalization can be used to improve performance for a variety of NLP tasks, including POS tagging (Derczynski et al., 2013), parsing (Zhang et al., 2013; Bhat et al., 2018), machine translation (Rosales Núñez et al., 2019), and named entity tagging (Schulz et al., 2016). Lexical normalization systems and benchmarks are available for multiple languages, we refer to Sharf and Rahman (2017) and van der Goot (2019) for an overview of available datasets and annotation efforts. Even though there is a rich stream of work on other natural language processing tasks for Italian social media (Bosco et al., 2016; Basile et al., 2016; Ronzano et al., 2018; Bosco et al., 2018), for lexical normalization only a rule-based system with a small test set is reported in the literature (Weber and Zhekova, 2016).

Original text							
joker	cmq	nn	e'	nnt	di	ke	!
Normalized version							
Joker	comunque	non	è	niente	di	che	!
English							
Joker	actually	not	is	worthwhile			!
Joker is actually not worthwhile !							

Figure 1: Example of a normalized sentence in Italian.

In this work, our main contributions are:

- The annotation of a lexical normalization dataset for Italian social media data containing 12,822 words;
- An analysis of the inter-annotator agreement for the task of lexical normalization;
- The evaluation of an existing lexical normalization model on this dataset;
- The addition of capitalization functionality in this normalization model and its evaluation.

[1] https://w3techs.com/technologies/history_overview/content_language

[2] Estimation based on tweets obtained by the random streaming API.

2 Related Work

Lexical normalization is considered to be beneficial for numerous tasks involving social media texts. Indeed, it has been successfully applied to POS tagging in English (Derczynski et al., 2013), Slovenian (Ljubešić et al., 2017), and Dutch (Schulz et al., 2016). When it comes to more complex tasks such as dependency parsing, the only experiments are for English (Zhang et al., 2013; van der Goot and van Noord, 2018).

A common shortcoming of lexical normalization datasets is their lack of information for the inter-annotator agreement. In literature, the only two datasets with an inter-annotator agreement study are from Pennell and Liu (2014) and Baldwin et al. (2015), both for English. Both works report kappa scores, with Pennell and Liu (2014) including the frequencies of how many annotators agree. In both cases, the kappa is calculated on the task of detecting whether a word should be normalized (because the number of classes/words is too large to calculate kappa scores for the full task).[3] Baldwin et al. (2015) report a Cohen's kappa (Cohen, 1960) of 0.5854, whereas Pennell and Liu (2014) report a Fleiss' kappa (Fleiss, 1971) of 0.891. This is a rather large difference, and it is unclear why this is the case. Possible explanations include differences in annotators, annotation guidelines, or data collection.

To get a clearer view on the agreement for the full task, we took the data from Pennell and Liu (2014), which includes annotation from multiple annotators, and we used it to calculate the agreement on the choice of the normalization in cases where annotators agree that the word needs to be replaced. The results show an observed agreement of 98.73%, leading to the conclusion that this last part of the task is easier.

Like most natural language processing tasks, most previous work on lexical normalization has focused on English. A wide variety of approaches is used; including machine translation (Aw et al., 2006), adapted spelling correction systems (Han and Baldwin, 2011), feature based systems (van der Goot, 2019; Jin, 2015), sequence to sequence models (Lourentzou et al., 2019), and recently contextualized embeddings (Muller et al., 2019). In this paper, we will make use of MoNoise (van der Goot, 2019), because it currently holds the state-of-art performance for multiple languages; it is described in more detail in Section 5.1.

The only previous attempt at normalizing Italian social media data is from Weber and Zhekova (2016). However, they have a different scope of the task, mostly focusing on readability, not on normalization on the *lexical* level. Besides spelling correction, their system also aims to remove words that do not contribute to the syntax, e.g. hashtags, emoticons, hyperlinks and other non-words. For this task, they propose a rule-based method, with a specific module for each type of task that they tackle.

In this work, we will use an existing Twitter dataset for Italian (Sanguinetti et al., 2018), to which we add a normalization annotation layer. It contains both random tweets and tweets related to politics and it has been annotated with different layers of information such as sentiment (Basile et al., 2014), POS tags (Bosco et al., 2016), and dependency structures (Sanguinetti et al., 2018).

[3] We confirmed this with the authors of Baldwin et al. (2015).

	Train/dev.	test
tweets	593	100
words	12,229	1,922
% OOV	34.70	29.5

Table 1: Some basic statistics of the filtered dataset.

3 Data

We annotated a subset of the data from Sanguinetti et al. (2018) (version 2.1). This dataset consists of 3,510 tweets and is a sub-set of two previously released datasets: SENTIPOLC (Barbieri et al., 2016) and SentiTUT (Bosco et al., 2013). Most of the tweets are collected in 2011 and 2012, and are filtered based on keywords on politics. There are some tweets from an earlier period (i.e., 2004) and a small sub-set is from the random Twitter API stream.

To ensure a basic annotation density and save time, we filtered the tweets which contain at least 3 out-of-vocabulary (OOV) words[4] to mainly focus on tweets containing non-standard language. A token is considered OOV if it does not appear in the Aspell[5] dictionary for Italian, or if it is either a url, a username, an hashtag, or if it only consists of punctuation. These latter elements have been identified by means of regular expressions. Furthermore, we created a small list of proper nouns from the most frequent OOV words in this dataset, and added them to the vocabulary. We maintained the splits of the original dataset (Sanguinetti et al., 2018). Because of the small size of the resulting data, we merge the training and development data, and perform experiments in a 10-fold setting. Basic statistics of the data after filtering are shown in Table 1.

4 Annotation

Annotation was done by four native speakers of Italian.[6] They are all male, between the age of 20 and 38 and from a variety of regions (Veneto, Tuscany, Liguria, and Apulia). All annotators have a background in natural language processing and are familiar with the Twitter platform.

The annotation has been conducted in two different steps and moments in time: first, the annotators were provided with no specific annotation guidelines but a description of the task and some examples in English. Each annotator annotated 50 tweets, resulting in 100 tweets which were marked up by two annotators, which will be our test data. For difficult cases, the annotators were encouraged to consult the Internet.[7] After this first round, annotation agreement was computed and the results were used to develop language specific annotation guidelines for Italian. These guidelines where then used to re-annotate the annotation of the 100 tweets annotated earlier as well as the training data.

[4] We supplemented the test data with 30 tweets with 2 OOV words to make it larger.

[5] `http://aspell.net/`

[6] The four last authors of this paper.

[7] Urban Dictionary, Google, and Twitter were used.

In the following sections, we present and discuss the guidelines and the agreement scores.

4.1 Guidelines

Baldwin et al. (2015)'s guidelines have been used as a starting point. In general, we correct similar phenomena, like character repetitions, typos, and word shortenings. In the following paragraphs, we illustrate specific instructions developed for Italian and other divergences from these guidelines. It should be noted that we did specifically focus on the extrinsic task of dependency parsing, but rather aim for a general lexical normalization model. Instructions are accompanied by examples showing how to correctly normalize each case.

Diacritics On social media, it is common to not type diacritics at all, inverting acute and grave accents, as well as to type the apostrophe next to the character. We correct all of these cases;

e' ↦ è é ↦ è perche' ↦ perché perchè ↦ perché

Capitalization In contrast to most previous work on other languages, we correct capitalization, including the first word of the sentence and named entities. It is worth noting that in Italian the use of capital letters beyond proper nouns is a complex topic. Indeed, some rules have been devised over time, for which concrete, unique entities must be capitalized. However, the "entification" is often a subjective process without clear boundaries, that is driven by the ideology, worship and psychological and linguistic disposition of the individuals.[8] Considering all this, we leave it up to the annotator whether a word should be capitalized, and in case of doubt, the original word should be kept. Some examples of corrected capitalization;

cisl ↦ CISL ROMPERE ↦ rompere
marc ↦ Marc AUGURI ↦ auguri

Dialects If there is a lexical equivalent in standard language, we choose to replace the word. Otherwise, we keep the word as is;

Terù ↦ Terrone freschìn ↦ freschìn
COJONI ↦ coglioni strafànto ↦ strafànto

Splitting of Tokens Sometimes words are incorrectly split or incorrectly merged into one word. We correct these cases. Merging only occurred once in our data, but splitting was more common;

Vabbene ↦ va bene

Contraction of Determiners A determiner can be contracted with an apostrophe (if the following noun is feminine) or without it (if the following noun is masculine). If there is a mismatch between the gender of the noun and the determiner, it is corrected, but otherwise contractions are kept;

un ↦ un un' ↦ un' un ultima ↦ un'ultima
dell' ↦ dell' l' ↦ l' un occhiata ↦ un'occhiata

Phrasal Abbreviations We exclude phrasal abbreviations from the annotation, because the written-out form does not correspond to the intended meaning of the phrase;

omg ↦ omg lol ↦ lol

Non-Words Interjections and non-words are excluded from annotation, as it is unclear what their normal form would be;

ahahah ↦ ahahah Bhè ↦ Bhè

Hashtags We keep usernames and hashtags as is, even if they are miss-spelled or contain multiple words;

#siamonoi ↦ #siamonoi #OroRosso ↦ #OroRosso
#piazzapulita ↦ #piazzapulita

Personal Pronouns and Clitic Pro-Forms Some words could be used both as pronouns or clitics. In the first round of annotation, we noticed some annotators had been more conservative than others, changing clitic forms (e.g., mi, ci) in their correspective stressed form (e.g., a me, a noi). Since clitic forms are part of the language, we decided to keep them;

mi ↦ mi (instead of "a me")
ci ↦ ci (instead of "a noi")
arrendermi ↦ arrendermi (instead of "arrendere me")
mouverci ↦ muoverci (instead of muovere "a noi")

Contraction of Prepositions In Italian, simple prepositions and definite articles are usually merged together. In syntactic annotation it is common to split those; however, since we aim for a general (not syntactically focused) normalization, and the contracted form is considered to be standard language,[9] we keep them;

del ↦ del (instead of "di il")
della ↦ della (instead of "di la")
sull' ↦ del (instead of "su l'")
alla ↦ alla (instead of "a la")

4.2 Agreement

In the first round of annotation, we used 100 random tweets from the test data. Annotators marked 50 random tweets each, resulting in 2 sets of annotations per tweet. Since normalization consists of two parts, i.e., (i) decide whether a word should be normalized; and (ii) choose the right replacement candidate, we computed the inter-annotator agreement on both aspects separately.
As already stated, the first round of annotation was conducted with no specific annotation guidelines. As this situation is a worst-case scenario, we also included agreement after some rule-based corrections for common clitics and capitalization categories (Section 4.1), as two annotators did not correct capitalization in the first round. More precisely, we always lower-cased tweets which are typed fully

[8] https://accademiadellacrusca.it/it/consulenza/uso-delle-maiuscole-e-minuscole/58

[9] http://www.grammatica-italiana.it/preposizioni-articolate.html

Metric	Before corrections	After corrections
Cohen's kappa	63.97	78.64
Word choice acc.	73.91	77.78

Table 2: Inter-annotator agreement for annotating lexical normalization without specific guidelines.

in uppercase and we ignored the contraction of determiners and clitics ("l"', "dell"', and "mi").

Inter-annotator agreements for both scenarios, i.e., before and after rule-based corrections, are reported in Table 2. Agreement is computed using Cohen's Kappa (Cohen, 1960). In particular, we converted the normalization annotations to a list of binary decisions (e.g., normalize the token or not). On top of this, we calculate the percentage of words for which annotators agreed on the normalization when they both agreed that the word is in need of normalization (i.e., word choice accuracy).

The Kappa scores are remarkably high considering the setup. In contrast to previous annotation for English (Pennell and Liu, 2014), there is a relatively large disagreement on the word choice task. However, upon inspection, we realized that most of these are capitalization issues. Indeed, even after our rule-based corrections there were inconsistencies for proper nouns (ignoring capitalization leads to an accuracy of 89%), which were unlikely to be real disagreements.

5 Lexical Normalization Model

In this section, we will evaluate an existing lexical normalization model, and extend it to handle capitalization better. We will first describe the model and data we used, then we evaluate and analyze it on the training data using 10-fold cross-validation. Finally, we will evaluate the model on the test data.

5.1 Model

We use MoNoise (van der Goot, 2019), because it reaches state-of-the-art performance on multiple languages and is publicly available. We report the results of a 10-fold cross-validation experiment on the training data and the results on the test data using the full training set. The model is using n-grams from an Italian Wikipedia dump from October 2019, and Twitter n-grams and skip-gram embeddings (Mikolov et al., 2013) based on Twitter data collected from the random Twitter stream from 2012 and 2018, filtered with the fastText language classifier (Joulin et al., 2017).[10]

5.2 Evaluation

Results of MoNoise on the training data are shown in Table 3. We report accuracy on the word level (including all words), and Error Reduction Rate (ERR) (van der Goot, 2019). We use a baseline which simply copies the original word (accuracy is equal to the ratio of words which are not changed in the annotation). In the top part of the table, we show the results when lowercase both the input data and the normalization annotation; this is a common setting for other benchmarks for the lexical normalization task.

	Accuracy	ERR
Lowercased		
base	97.13	0.0
MoNoise	97.83	32.00
Not lowercased		
base	92.61	0.0
MoNoise	94.07	24.67
+capFeats	94.93	45.87

Table 3: Results of MoNoise on the training data with 10-fold cross validation. +capFeats: MoNoise including features to improve capitalization handling.

When evaluating MoNoise without lowercasing all data (bottom part of Table 3), the ERR is higher. This is surprising, because the model is never evaluated for this exact task. However, when manually looking at the errors, we found that the model still made a lot of errors in capitalization (accounting for approximately 5 percentage points in accuracy). For this reason we added simple features to MoNoise. First, we added a generation module which adds a lowercased version of the original word, and a copy where only the first character is uppercase. Second, we add a feature which indicates the index of the word in the sentence. Because in Italian the first word of the sentence should be capitalized, this feature should be informative. After these additions, we can see a performance boost of more than 20 points in ERR (+capFeats in Table 3).

To test whether the fully automatic approach has similar difficulties as the human annotators, we checked the performance of the model on the detection of words in need of normalization. The results on this task are only slightly higher compared to the full normalization task, indicating that the model has similar difficulties. In general, it is too conservative in replacing words (in 90% of the mistakes, it kept the original word, in 10% it normalized too aggressively). This means that the model prefers precision over recall, which arguably is a desirable setting.

5.3 Test Data

On the test data, we only run the capitalization-enabled version of MoNoise to encourage future work to also include capitalization during evaluation. The results of the test data (Table 4) show that the results are even lower on this particular split. This is perhaps an effect of having more noise, as 11.49% of the words are changed versus 9.77% on the development data. In contrast to the development data, results are much lower compared to other languages for which MoNoise is evaluated (van der Goot, 2019). When inspecting the different folds, it becomes clear that the performance is somewhat unstable. On eight folds, the ERR is higher than 62, whereas on one fold the ERR is only 39 (standard deviation is 8). This is most likely an effect of the limited amount of training data.

[10]The complete model including n-gram frequencies and word embeddings is published at `www.robvandergoot.com/data/monoise/it.tar.gz`

Metric	Baseline	MoNoise
Accuracy	94.89	96.74
ERR	0.0	36.27

Table 4: Results of MoNoise on the test data compared to the baseline.

6 Applicability for Dependency Parsing

To evaluate the effect of normalization on dependency parsing, we use the UUParser 2.3 (Smith et al., 2018) with default settings. We choose this parser because it is easy and fast to train, and it performs well on tweets.[11] We evaluate two settings, one where we train the dependency on social media data, and a domain adaptation setup where we use canonical data from the ISDT treebank (Bosco et al., 2014). For the setting with Twitter data we only use data where normalization annotation is available (i.e. we use exactly the same data as in Section 5) to be able to also run the full experiment with gold-normalized training data.[12] It should be noted that the tokenization of the normalization is different compared to the treebank data; clitics and prepositions are split in the treebank (Section 4.1). This potentially has a large effect on the performance, while the splitting is trivial (it is a closed class of cases) so we decided to use the gold splitting during training and testing.

Similar to Section 5 we report average scores of 10 folds on the training data in the Twitter setting. For this setup, we will examine three settings: (1) baseline setting which makes no use of normalization, (2) gold normalization, and (3) predicted normalization. Each of these settings is applied to both the development and the training splits. We report results of all combinations.

We use LAS as defined by Zeman et al. (2018) for evaluation. Results are shown in Table 5. When not normalizing the training data (first row), gold normalization only leads to an performance improvement of 0.44 LAS points and predicted normalization scores even slightly higher. When normalizing the training data, it becomes apparent that the normalization strategy (gold/predicted) should be the same as on the development data for optimal results. Overall, using gold normalization a performance increase of 1.47 LAS can be obtained, whereas with predicted normalization, the improvement is only 0.71 LAS point.

On the test data, the parsers trained on canonical data (ISDT) initially scores higher, in contrast to the results on the train data. This suggests that the syntax of this split is more similar to the syntax of standard language. As an effect of this, the gains from using normalization are also smaller for the ISDT parser compared to the parser trained on tweets. When using the Twitter based parser, the gold normalization still shows a relatively large gap compared to the performance of predicted normalization. However, perhaps surprisingly, when training on canonical data (ISDT), using predicted normalization on the input data leads to a slightly better performance compared to using gold. However, the differences are very minor in this setting, and considering the size of the test data (100 tweets), we can not draw any conclusions from these results.

[11] The UUParser scored third on the full treebank in the CoNLL 2018 Shared Task (Zeman et al., 2018).

[12] Our results are thus not directly comparable to previous work, as we use different training data.

	Development		
Train	Base	Pred.	Gold
Twitter.base	67.11	67.60	67.55
Twitter.pred	66.49	67.82	67.79
Twitter.gold	66.49	67.98	68.23
ISDT	62.70	64.37	64.58

Table 5: LAS of dependency parser trained on different alternations of the data.

Train	Base	Pred.	Gold
Twitter	60.86	62.27	63.72
ISDT	65.96	66.29	66.10

Table 6: Parser results on the test data (LAS). In the 'Twitter' setting the same normalization strategy is used for train and test data.

7 Conclusion

We introduced a novel benchmark for lexical normalization of Italian social media data. We first used this benchmark to get a more detailed understanding of inter-annotator agreement for this task, and learned that the most difficulties are found in the task of error detection, i.e. the decision whether a word requires normalization or not. Furthermore, we utilized this dataset to train a lexical normalization model (MoNoise) and evaluated performance. It became clear that this is a difficult task, and scores are lower than for most other (Indo-European) languages, using the same model. To improve performance for capitalization corrections, we added a generation module and a feature, and showed that this leads to a boost in performance. Furthermore, we saw that the mistakes of the normalization model are similar to the disagreements of the human annotators (deciding when to normalize). Finally, we showed that this normalization model can be used to improve dependency parsing.

8 Bibliographical References

Aw, A., Zhang, M., Xiao, J., and Su, J. (2006). A phrase-based statistical model for sms text normalization. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 33–40. Association for Computational Linguistics.

Baldwin, T., de Marneffe, M.-C., Han, B., Kim, Y.-B., Ritter, A., and Xu, W. (2015). Shared Tasks of the 2015 Workshop on Noisy User-generated Text: Twitter Lexical Normalization and Named Entity Recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135, Beijing, China, July. Association for Computational Linguistics.

Barbieri, F., Basile, V., Croce, D., Nissim, M., Novielli, N., and Patti, V. (2016). Overview of the evalita 2016

sentiment polarity classification task. In *Proceedings of Evalita 2016*.

Basile, V., Bolioli, A., Nissim, M., Patti, V., and Rosso, P. (2014). Overview of the Evalita 2014 SENTIment POLarity Classification Task. In *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 & and of the Fourth International Workshop EVALITA 2014*, pages 50–57. Pisa University Press.

Basile, P., Caputo, A., Gentile, A. L., and Rizzo, G. (2016). Overview of the EVALITA 2016 named entity recognition and linking in Italian tweets (NEEL-IT) task. In *Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016*, page 40.

Bhat, I., Bhat, R. A., Shrivastava, M., and Sharma, D. (2018). Universal dependency parsing for Hindi-English code-switching. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 987–998, New Orleans, Louisiana, June. Association for Computational Linguistics.

Bosco, C., Patti, V., and Bolioli, A. (2013). Developing corpora for sentiment analysis: The case of irony and Senti-TUT. *IEEE Intelligent Systems*, 28(2):55–63.

Bosco, C., Dell'Orletta, F., Montemagni, S., Sanguinetti, M., and Simi, M. (2014). The Evalita 2014 Dependency Parsing task. In *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 & and of the Fourth International Workshop EVALITA 2014*, pages 1–8. Pisa University Press.

Bosco, C., Fabio, T., Andrea, B., and Mazzei, A. (2016). Overview of the Evalita 2016 part of speech on Twitter for Italian task. In *Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, volume 1749, pages 1–7. CEUR Workshop Proceedings (CEUR-WS. org).

Bosco, C., Felice, D., Poletto, F., Sanguinetti, M., and Maurizio, T. (2018). Overview of the EVALITA 2018 Hate Speech Detection Task. In *EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*, volume 2263, pages 1–9. CEUR.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Derczynski, L., Ritter, A., Clark, S., and Bontcheva, K. (2013). Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 198–206, Hissar, Bulgaria, September. INCOMA Ltd. Shoumen, BULGARIA.

Eisenstein, J. (2013). What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, Georgia, June. Association for Computational Linguistics.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5).

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 42–47, Portland, Oregon, USA, June. Association for Computational Linguistics.

Han, B. and Baldwin, T. (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.

Jin, N. (2015). NCSU-SAS-ning: Candidate generation and feature engineering for supervised lexical normalization. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 87–92, Beijing, China, July. Association for Computational Linguistics.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics, April.

Kiperwasser, E. and Goldberg, Y. (2016). Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Liu, Y., Zhu, Y., Che, W., Qin, B., Schneider, N., and Smith, N. A. (2018). Parsing tweets into universal dependencies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 965–975, New Orleans, Louisiana, June. Association for Computational Linguistics.

Ljubešić, N., Erjavec, T., and Fišer, D. (2017). Adapting a state-of-the-art tagger for south Slavic languages to non-standard text. In *Proceedings of the 6th Workshop on Balto-Slavic natural language processing*, pages 60–68.

Lourentzou, I., Manghnani, K., and Zhai, C. (2019). Adapting sequence to sequence models for text normalization in social media. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 335–345.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*.

Muller, B., Sagot, B., and Seddah, D. (2019). Enhancing BERT for lexical normalization. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 297–306, Hong Kong, China, November. Association for Computational Linguistics.

Pennell, D. L. and Liu, Y. (2014). Normalization of informal text. *Computer Speech & Language*, 28(1):256–277.

Plank, B. (2016). What to do about non-standard (or non-

canonical) language in NLP. *Proceedings of the 13th Conference on Natural Language Processing (KONVENS)*.

Ronzano, F., Barbieri, F., Wahyu Pamungkas, E., Patti, V., Chiusaroli, F., et al. (2018). Overview of the EVALITA 2018 Italian Emoji Prediction (ITAMoji) Task. In *6th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop, EVALITA 2018*, volume 2263, pages 1–9. CEUR-WS.

Rosales Núñez, J. C., Seddah, D., and Wisniewski, G. (2019). Phonetic normalization for machine translation of user generated content. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 407–416, Hong Kong, China, November. Association for Computational Linguistics.

Sanguinetti, M., Bosco, C., Lavelli, A., Mazzei, A., Antonelli, O., and Tamburini, F. (2018). PoSTWITA-UD: an Italian Twitter Treebank in universal dependencies. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan, May. European Languages Resources Association (ELRA).

Schulz, S., Pauw, G. D., Clercq, O. D., Desmet, B., Hoste, V., Daelemans, W., and Macken, L. (2016). Multimodular Text Normalization of Dutch User-Generated Content. *ACM Transactions on Intelligent Systems Technology*, 7(4):1–22, July.

Sharf, Z. and Rahman, S. U. (2017). Lexical normalization of roman Urdu text. *International Journal of Computer Science and Network Security*, 17(12):213–221.

Smith, A., Bohnet, B., de Lhoneux, M., Nivre, J., Shao, Y., and Stymne, S. (2018). 82 Treebanks, 34 Models: Universal Dependency Parsing with Multi-Treebank Models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium, October. Association for Computational Linguistics.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 252–259.

van der Goot, R. and van Noord, G. (2018). Modeling input uncertainty in neural network dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4984–4991. Association for Computational Linguistics.

van der Goot, R. (2019). MoNoise: A Multi-lingual and Easy-to-use Lexical Normalization Tool. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 201–206, Florence, Italy, July. Association for Computational Linguistics.

Weber, D. and Zhekova, D. (2016). TweetNorm: Text Normalization on Italian Twitter Data. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, pages 306–312.

Zeman, D., Hajič, J., Popel, M., Potthast, M., Straka, M., Ginter, F., Nivre, J., and Petrov, S. (2018). CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium, October. Association for Computational Linguistics.

Zhang, C., Baldwin, T., Ho, H., Kimelfeld, B., and Li, Y. (2013). Adaptive Parser-Centric Text Normalization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1159–1168, Sofia, Bulgaria, August. Association for Computational Linguistics.