An In-depth Analysis of the Effect of Text Normalization in Social Media

Tyler Baldwin*
baldwin.tyler.s@gmail.com

Yunyao Li
IBM Research - Almaden
650 Harry Road
San Jose, CA 95120, USA
yunyaoli@us.ibm.com

Abstract

Recent years have seen increased interest in text normalization in social media, as the informal writing styles found in Twitter and other social media data often cause problems for NLP applications. Unfortunately, most current approaches narrowly regard the normalization task as a "one size fits all" task of replacing non-standard words with their standard counterparts. In this work we build a taxonomy of normalization edits and present a study of normalization to examine its effect on three different downstream applications (dependency parsing, named entity recognition, and text-to-speech synthesis). The results suggest that how the normalization task should be viewed is highly dependent on the targeted application. The results also show that normalization must be thought of as more than word replacement in order to produce results comparable to those seen on clean text.

1 Introduction

The informal writing style employed by authors of social media data is problematic for many natural language processing (NLP) tools, which are generally trained on clean, formal text such as newswire data. One possible solution to this problem is normalization, in which the informal text is converted into a more standard formal form. Because of this, the rise of social media data has coincided with a rise in interest in the normalization problem.

Unfortunately, while many approaches to the problem exist, there are notable limitations to the way in which normalization is examined. First, although social media normalization is universally motivated by pointing to its role in helping downstream applications, most normalization work gives little to no insight into the effect of the normalization process on the downstream application of interest. Further, the normalization process is generally seen to be agnostic of the downstream application, adopting a "one size fits all" view of how normalization should be performed. This view seems intuitively problematic, as different information is likely to be of importance for different tasks. For instance, while capitalization is important for resolving named entities, it is less important for other tasks, such as dependency parsing.

Some recent work has given credence to the idea that application-targeted normalization is appropriate (Wang and Ng, 2013; Zhang et al., 2013). However, how certain normalization actions influence the overall performance of these applications is not well understood. To address this, we design a taxonomy of possible normalization edits based on inspiration from previous work and an examination of annotated data. We then use this taxonomy to examine the importance of individual normalization actions on three different downstream applications: dependency parsing, named entity recognition, and text-to-speech synthesis. The results suggest that the importance of a given normalization edit is highly dependent on the task, making the "one size fits all" approach inappropriate. The results also show that a narrow view of normalization as word replacement is insufficient, as many often-ignored normalization actions prove to be important for certain tasks.

* Work was done while at IBM Research - Almaden.

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 420–429,
Denver, Colorado, May 31 – June 5, 2015.

In the next section, we give an overview of previous work on the normalization problem. We then introduce our taxonomy of normalization edits in Section 3. In Section 4, we present our evaluation methodology and present results over the three applications, using Twitter data as a representative domain. Finally, we discuss our results in Section 5 and conclude in Section 6.

2 Related Work

Twitter and other social media data is littered with non-standard word forms and other informal usage patterns, making it difficult for many NLP tools to produce results comparable to what is seen on formal datasets. There are two approaches proposed in the literature to handle this problem (Eisenstein, 2013). One approach is to tailor a specific NLP tool towards the data, by using training data from the domain to help the tool learn its specific idiosyncrasies. This approach has been applied with reasonable success on named entity recognition (Liu et al., 2011b; Ritter et al., 2011) as well as on parsing and part-of-speech tagging (Foster et al., 2011).

The other approach is normalization. Rather than tailoring a NLP tool towards the data, normalization seeks to tailor the data towards the tool. This is accomplished by transforming the data into a form more akin to the formal text that NLP tools are generally trained on. While normalization is often more straightforward and more easily applied in instances in which retraining is difficult or impractical, it has potential disadvantages as well, such as the potential loss of pragmatic nuance (Baldwin and Chai, 2011).

Prior to the rise of social media, the normalization process was primarily seen as one of standardizing non-standard tokens found in otherwise clean text, such as numbers, dates, and acronyms (Sproat et al., 2001). However, the current popularity of Twitter and other informal texts has caused the normalization task to take on a broader meaning in these contexts, where the goal is to convert informal text into formal text that downstream applications expect.

Many different approaches to social media normalization have been undertaken. These approaches often draw inspiration from other tasks such as machine translation (Pennell and Liu, 2011; Aw et al., 2006), spell checking (Choudhury et al., 2007) or speech recognition (Kobus et al., 2008). Other approaches include creating automatic abbreviations via a maximum entropy classifier (Pennell and Liu, 2010), creating word association graphs (Sonmez and Ozgur, 2014), and incorporating both rules and statistical models (Beaufort et al., 2010). While most initial approaches used supervised methods, unsupervised methods have recently become popular (Cook and Stevenson, 2009; Liu et al., 2011a; Yang and Eisenstein, 2013; Li and Liu, 2014). Some work has chosen to focus on specific aspects of the normalization process, such as providing good coverage (Liu et al., 2012) or building normalization dictionaries (Han et al., 2012).

In all of the work mentioned above, the normalization task was seen primarily as one of converting non-standard tokens into an equivalent standard form. Similarly, many of these works defined the problem even more narrowly such that punctuation, capitalization, and multi-word replacements were ignored. However, two pieces of recent work have suggested that this understanding of the normalization task is too narrow, as it ignores many other hallmarks of informal writing that are prevalent in social media data. Wang and Ng (2013) present a beam search based approach designed to handle machine translation which incorporates attempts to correct mistaken punctuation and add missing words, such as forms of the verb *to be*. Similarly, Zhang et al. (2013) attempt to perform all actions necessary to create a formal text. In both instances the work was motivated by, and evaluated with respect to, a specific downstream application (machine translation and dependency parsing, respectively). However, not every study that tied the output to an application chose a broad interpretation of the normalization problem (Beaufort et al., 2010; Kaji and Kitsuregawa, 2014).

3 Taxonomy of Normalization Edits

In order to understand the impact of individual normalization edits on downstream applications, we first need to define the space of possible normalization edits. While it is not uncommon for normalization work to present some analysis of the data, these analyses are often quite specific to the domain and datasets of interest. Because there is no agreed upon

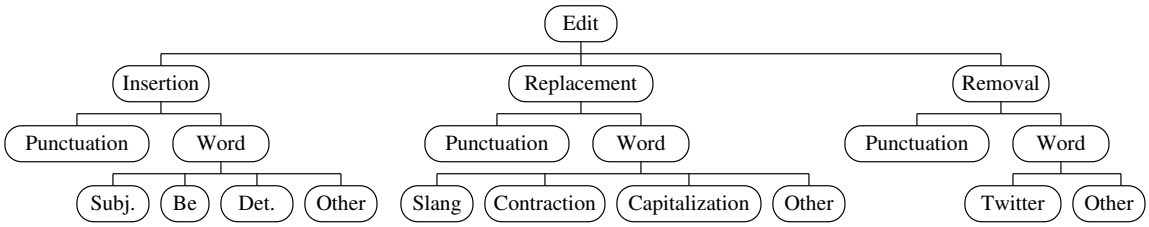

Figure 1: Taxonomy of normalization edits

taxonomy of normalization token or edit types, different analyses often look at different edit types and at different levels of granularity. In an attempt to help future work converge on a common understanding of normalization edits, in this section we present our taxonomy of normalization edits at several different levels of granularity. While it would be difficult for a taxonomy of normalization edits to be universal enough to be appropriate over all datasets and domains, we attempt to provide a taxonomy general enough to give future work a meaningful initial point of reference.

3.1 Methodology

Our taxonomy draws inspiration from both previous work and an examination of our own dataset (Section 3.3). In doing so, it attempts to cover normalization edits broadly, including cases that are universally understood to be important, such as slang replacement, as well as cases that are frequently ignored, such as capitalization correction.

One of the guiding principles in the design of our taxonomy was that categories should not be divided so narrowly such that the phenomenon they capture appeared very infrequently in the data. One example of this is our decision not to divide punctuation edits at the lowest level of granularity. While certain clear categories exist (e.g., emoticons), these cases appeared in a small enough percentage of tokens that they would be difficult to examine and likely have a negligible effect on overall performance.

3.2 Taxonomy

Our taxonomy of normalization edits is shown in Figure 1. As can be seen, we categorize edits at three levels of granularity.

Level One. The primary goal of the level one segmentation is to separate token replacements which are most centrally thought of as part of the normalization task from other instances that may require additional pragmatic inference. Specifically, we separate edits coarsely into three categories:

- **Token Replacements.** Replacing one or more existing tokens with one or more new tokens (e.g., replacing *wanna* with *want to*).
- **Token Additions.** Adding a token that does not replace an existing token (e.g., adding in missing subjects).
- **Token Removals.** Removing a token without replacing it with an equivalent (e.g., removing laughter words such as *lol* and *hahaha*).

Level Two. The next level of granularity separates normalization edits over word tokens from those over punctuation:

- **Word.** Replacing, adding, or removing word tokens (depending on parent).
- **Punctuation.** Replacing, adding, or removing punctuation tokens (depending on parent).

Level Three. At the final level, we subdivide word edits into groups as appropriate for the edit type. Rather than attempting to keep consistent groups across all leaf nodes, we selected the grouping based on the data distribution. For instance, Twitter-specific tokens (e.g., retweets) are often removed during normalization, so examining the removal of these words as a group is warranted. In contrast, these tokens are never added, so different segmentation is appropriate when examining word addition.

At the lowest level of the taxonomy, word replacements were subdivided as follows:

- **Contraction Replacements.** Unrolling standard contractions (*don't*), common informal cases (*wanna*), and non-standard variations produced via apostrophe omission (*dont*).

- **Slang Replacements.** Replacing slang terms, such as slang shortenings and word elongation.

- **Capitalization Replacements.** Correcting the capitalization of words. The replaced word differs from its replacement by *only* capitalization.

- **Other Replacements.** Correcting unintentional typographic mistakes, such as misspelling and word concatenation.

When segmenting word additions, we note that words that need to be added in a normalization edit were often consciously dropped by the user in the original text. Our categorization reflects this by examining syntactic categories that are often dropped in informal writing:

- **Subject Addition.** Adding in omitted subjects.

- **Determiner Addition.** Adding in omitted determiners (e.g., "[*The*] front row is so close").

- **Be-verb Addition.** Adding in omitted forms of the verb *to be*.

- **Other Addition.** All word additions not covered by the other categories.

Finally, word removals are subdivided into just two categories:

- **Dataset-specific Removals.** Removing tokens that do not appear outside of the dataset in question (e.g., for Twitter: hashtags, @replies, and retweets).

- **Other Removals.** Removing interjections, laughter words, and other expression of emotion (e.g., *ugh*).

Note that we are not suggesting here that dataset-specific words should be removed in all cases. While in many cases they may be removed if they do not have a formal equivalent, they may also be replaced or retained as is, depending on the context.

3.3 Dataset

To facilitate our experiments, we collected and annotated a dataset of Twitter posts (tweets) from the TREC Twitter Corpus[1]. The TREC Twitter corpus is a collection of 16 million tweets posted in January and February of 2011. The corpus is designed to be a representative sample of Twitter usage, and as such includes both regular and spam tweets. To build our dataset, we sampled 600 posts at random from the corpus. The tweets were then manually filtered such that tweets that were not in English were replaced with those in English.

To produce our gold standard, two oDesk[2] contractors were asked to manually normalize each tweet in the dataset to its fully grammatical form, as would be found in formal text. Annotation guidelines stipulated that twitter-specific tokens should be retained if important to understanding the sentence, but modified or removed otherwise. As noted, most previous work often stopped short of requiring full grammaticality. However, Zhang et al. (2013) argued that grammaticality should be the ideal end goal of normalization since the models used in downstream applications are typically trained on well-formed sentences. We adopt this methodology here both because we agree with this assertion and because a fully grammatical form is appropriate for all of the downstream applications of interest, allowing for a single unified gold standard that can aid comparison across applications.

During gold standard creation, each normalization edit was labeled with its type, according to the above taxonomy. The distribution of normalization edits in the dataset is given in Table 1. As shown, normalization edits accounted for about 29% of all tokens. Token replacements accounted for just over half of all edits (53%), while token addition (29%) was more common than token removal (18%). One interesting observation is non-capitalization word replacement accounted for only 25% of all normalization edits, intuitively indicating potential drawbacks for the common definition of normalization as one of simple word replacement which ignores capitalization and punctuation.

[1] http://trec.nist.gov/data/tweets/

[2] https://www.odesk.com/

Configuration	Count
No edit	8479
All edits	3411
ADDITION	993
PUNCT	437
WORD	556
BEVERB	137
DETERMINER	103
OTHER	141
SUBJECT	175
REPLACEMENT	1797
PUNCT	312
WORD	1485
CAPITALIZATION	634
CONTRACTION	246
OTHER	176
SLANG	429
REMOVAL	621
PUNCT	120
WORD	501
OTHER	172
TWITTER	329

Table 1: Token counts for each type of normalization edit.

4 Evaluation

In this section, we present our examination of the effect of normalization edits on downstream NLP applications. To get a broad understanding of these effects, we examine three very different cases: dependency parsing, named entity recognition (NER), and text-to-speech (TTS) synthesis. We chose these tasks because they each require the extraction of different information from the text. For instance, named entity recognition requires only a shallow syntactic analysis, in contrast to the deeper understanding required for dependency parsing. Similarly, only speech synthesis requires phoneme production, while the other tasks do not. Despite their differences, each of these tasks is relevant to larger applications that would benefit from improved performance on Twitter data, and each has garnered attention in the normalization and Twitter-adaptation literature (Beaufort et al., 2010; Liu et al., 2011b; Zhang et al., 2013).

Although the differences in these tasks also dictates that they be evaluated somewhat differently, we examine them within a common evaluation structure. In all cases, to examine the effects of each normalization edit we model our analyses as ablation studies. That is, for every category in the taxonomy, we examine the effect of performing all normalization edits *except* the relevant case. This allows us to measure the drop in performance solely attributable to each category; the greater the performance drop observed when a given normalization edit is not performed, the greater the importance of performing that edit.

To aid analysis, results are presented in two ways: 1) as raw performance numbers, and 2) as an error rate per-token. These metrics give two different views of the relevance of each edit type. The raw numbers give a sense of the overall impact of a given category, and as such may be impacted by the size of the category, with common edits becoming more important simply by virtue of their frequency. In contrast, the per-token error rate highlights the cost of failing to perform a single instance of a given normalization edit, independent of the frequency of the edit. Both of these measures are likely to be relevant when attempting to improve the performance of a normalization system. Note that since the first measure is one of overall performance, smaller numbers reflect larger performance drops when removing a given type of edit, so that the smaller the number the more critical the need to perform the given type of normalization. In contrast, the latter judgment is one of error rate, and thus interpretation is reversed; the larger the error rate when it is removed, the more critical the normalization edit.

Another commonality among the analyses is that performance is measured relative to the top performance of the *tool*, not the task. That is, following Zhang et al. (2013), we consider the output produced by the tool (e.g., the dependency parser) on the grammatically correct data to be gold standard performance. This means that some output based on our gold standard may in fact be incorrect relative to human judgment, simply because the tool used does not have perfect performance even if the text if fully grammatical. Since the goal is to understand how normalization edits impact the performance, this style of evaluation is appropriate; it considers mistakes attributable to normalization edits as erroneous, but ignores those mistakes attributable to the limitations of the tool.

Finally, to maximize the relevance of the analyses

given here, in each case we employ publicly available and widely used tools.

4.1 Parser Evaluation

To examine the effect of normalization on dependency parsing, we employ the Stanford dependency parser[3] (Marneffe et al., 2006). To produce the gold standard dependencies for comparison, the manually grammaticalized tweets (Section 3.3) were run through the parser. To compare the ablation results to the gold standard parses, we adopt a variation of the evaluation method used by Zhang et al. (2013). Given dependency parses from the gold standard and a candidate normalization, we define precision and recall as follows:

$$\text{precision}_{\text{sov}} = \frac{|SOV \cap SOV_{gold}|}{|SOV|} \quad (1)$$

$$\text{recall}_{\text{sov}} = \frac{|SOV \cap SOV_{gold}|}{|SOV_{gold}|} \quad (2)$$

Where SOV and SOV_{gold} are the sets of subject, object, and verb dependencies in the candidate normalization and gold standard, respectively. While Zhang et al. chose to examine subjects and objects separately from verbs, we employ a unified metric to simplify interpretation.

4.1.1 Results

Results of the ablation study are summarized in Table 2. As shown, the performance of a complex task such as dependency parsing is broadly impacted by a variety of normalization edits. Based on the raw F-measure, the more common word replacements proved to be the most critical, although failing to handle token addition and removal edits also resulted in substantial drops in performance. At the lowest level in the taxonomy, slang replacements and subject addition were the most critical edits.

Although many replacement tasks were important in aggregate, on a per-token basis the most important edits were those that required token removal and addition. Perhaps unsurprisingly, failing to add subjects and verbs resulted in the largest issues, as the parser has little chance of identifying these dependencies if the terms simply do not appear in the sentence. However, not all word additions proved critical, as failing to add in a missing determiner generally had little impact on the overall performance. Similarly, failing to correct capitalization did not cause substantial problems for the parser. Some word replacements did prove to be important, with slang and other word replacements showing some of the largest per-token error rates. Removing misleading punctuation or changing non-standard punctuation both proved important, but the per-token effect of punctuation addition was modest.

[3]Version 2.0.5

Configuration	F-measure	Per-token Error Rate
-ADDITION	0.790	0.00021
-PUNCT	0.919	0.00019
-WORD	0.842	0.00028
-BEVERB	0.948	0.00038
-DETERMINER	0.980	0.00019
-OTHER	0.959	0.00029
-SUBJECT	0.903	**0.00055**
-REPLACEMENT	**0.710**	0.00016
-PUNCT	0.907	0.00030
-WORD	**0.754**	0.00017
-CAPITALIZATION	0.950	0.00008
-CONTRACTION	0.945	0.00023
-OTHER	0.947	0.00030
-SLANG	**0.872**	0.00030
-REMOVAL	0.866	**0.00022**
-PUNCT	0.959	**0.00034**
-WORD	0.887	0.00023
-OTHER	0.952	0.00028
-TWITTER	0.925	0.00023

Table 2: Dependency Parser Results.

In general, the results suggest that a complex task such as dependency parsing suffers substantially when the input data differs from formal text in any number of ways. With the exception of capitalization correction, performing almost every normalization edit is necessary to achieve results commensurate with those seen on formal text.

4.2 NER Evaluation

In this section, we examine the effect of each normalization edit on a somewhat more shallow interpretation task, named entity recognition. Unlike dependency parsing which requires an understanding of every token in the text, NER must only determine whether a given token is a named entity, and if so, discover its associated entity type.

The setup for evaluation of normalization edits on named entity recognition closely follows that of dependency parsing. We once again employ a tool from the suite of Stanford NLP tools, the Stanford named entity recognizer[4] (Finkel et al., 2005). We also define precision and recall in a similar manner:

$$\text{precision}_{\text{ner}} = \frac{|ENT \cap ENT_{gold}|}{|ENT|} \quad (3)$$

$$\text{recall}_{\text{ner}} = \frac{|ENT \cap ENT_{gold}|}{|ENT_{gold}|} \quad (4)$$

Where ENT and ENT_{gold} are the sets of entities identified over the candidate normalization and gold standard sentences, respectively. Entities were labeled as one of three classes (`person`, `location`, or `organization`), and two entities were only considered a match if they both selected the same entity and the same entity class.

4.2.1 Results

Table 3 shows the results of the NER ablation study. Unlike dependency parsing, only word replacement edits proved to be critically important for NER tasks, as adding and subtracting words had little impact on the overall performance. Capitalization, which is generally an important feature for the identification of named entities, was unsurprisingly important. Similarly, the replacement of word types other than slang and contraction was important, because many of these instances may come from misspelled named entities. Slang and contractions were less important, as they were generally not used to reference named entities. As the words dropped by Twitter users tend to be function words that are not critical to understanding the sentence they are rarely named entities and have only a small effect on named entity recognition. Similarly, terms that are removed during normalization also tend to not be named entities, and thus has minor overall impact.

A similar phenomenon is observed in the per-token evaluation, where unintentionally produced, non-slang, non-contraction word replacement was seen to be of paramount importance. Punctuation removal was also important on a per-token basis, despite having little impact in aggregate.

Overall, the results given in Table 3 indicate that a focused approach to normalization for named entity recognition is warranted. Unlike dependency parsing that required a broad approach involving token addition and removal, the replacement-centric normalization approach typically employed by previous work is likely to be sufficient when the goal is to improve entity recognition.

[4]Version 1.2.8

Configuration	F-measure	Per-token Error Rate
-ADDITION	0.955	0.00005
-PUNCT	0.973	0.00006
-WORD	0.974	0.00005
-BEVERB	0.998	0.00001
-DETERMINER	0.989	0.00011
-OTHER	0.989	0.00008
-SUBJECT	0.998	0.00001
-REPLACEMENT	**0.827**	**0.00010**
-PUNCT	0.962	0.00012
-WORD	**0.849**	0.00010
-CAPITALIZATION	**0.921**	0.00012
-CONTRACTION	0.977	0.00009
-OTHER	0.931	**0.00039**
-SLANG	0.945	0.00013
-REMOVAL	0.956	0.00007
-PUNCT	0.970	**0.00025**
-WORD	0.960	0.00008
-OTHER	0.973	0.00015
-TWITTER	0.962	0.00012

Table 3: NER Results.

4.3 TTS Evaluation

Unlike the previous two tasks, the TTS problem is complicated by its need for speech production. Similarly, evaluation of speech synthesis is more difficult, as it requires human judgment about the overall quality of the output (Black and Tokuda, 2005). While speech synthesis evaluations often rate performance on a 5 point scale, we adopt a more restricted method, based on the comparison to gold standard methodology used in the previous evaluations. For each tweet and each round of ablation, a synthesized audio file was produced from both the gold standard and ablated version of the tweet. These audio snippets were randomized and presented to human judges who were asked to make a binary judgment as to whether the meaning and understandability of the ablated case was comparable to the gold standard. The accuracy of a given round of ablation is then calculated to be the percentage of tweets judged

Configuration	F-measure	Per-token Error Rate
-ADDITION	0.713	0.00029
-PUNCT	0.920	0.00018
-WORD	0.723	0.00050
-BEVERB	0.903	0.00071
-DETERMINER	0.937	0.00061
-OTHER	0.910	0.00064
-SUBJECT	0.853	0.00084
-REPLACEMENT	**0.550**	0.00025
-PUNCT	0.877	0.00040
-WORD	**0.590**	0.00028
-CAPITALIZATION	0.860	0.00022
-CONTRACTION	0.910	0.00037
-OTHER	0.883	0.00066
-SLANG	0.783	0.00051
-REMOVAL	0.580	**0.00068**
-PUNCT	0.880	**0.00100**
-WORD	0.600	0.00080
-OTHER	0.837	**0.00095**
-TWITTER	**0.710**	0.00088

Table 4: Text-To-Speech Synthesis Results.

to be similar to the gold standard.

The eSpeak speech synthesizer[5] was used to produce audio files for all tweet variations in the ablation study. As is common for speech synthesizers, eSpeak does perform some amount of TTS-specific normalization natively. While this does influence the normalizations produced, the comparison to gold standard methodology employed in this study helps us to focus on differences that are primarily attributable to the normalization edits we wish to examine, not those produced natively. To obtain the gold standard, two native-English speaking judges were recruited via oDesk. Inter-annotator agreement was moderate, $\kappa = 0.48$.

4.3.1 Results

Table 4 shows the results of the speech synthesis study. As shown, the removal of non-standard or out of place tokens was most critical to the production of a normalization that is clearly understandable to human listeners. The aggregate results for token removals were comparable to or better than those of replacements at all levels of the taxonomy, in contrast to the results from the other two tasks, where the larger number of replacements led to the largest performance hits. Meanwhile, word addition proved to be less essential overall.

At the token level, the importance of token removal is even more stark; the per-token error rate of every category of removal is greater than that of all other categories at the same taxonomy level. Although most word additions had a comparatively small effect on performance overall, they were important on a per-token basis. Most notably, subject adding had high per-token importance. In contrast, failing to add missing punctuation was not often marked as erroneous by human judges, nor was failing to normalize capitalization or contractions.

Similar to those on dependency parsing, the results on speech synthesis suggest that a broad approach that considers several different types of normalization edit is necessary to produce results comparable to those seen on clean text. However, at a high level there is a clear divide in importance between normalization types, where the greatest performance gains can be obtained by focusing on the comparatively small number of token removals.

5 Discussion

The results presented in Section 4 are consistent with the hypothesis that a "one size fits all" approach to Twitter normalization is problematic, as the importance of a given normalization edit was highly dependent on the intended downstream task. Differences in which edits had the most substantial effect were present at all levels of scrutiny. Adding subjects and other words that a Twitter author dropped can be vitally important if the goal is to improve parsing performance, but can mostly be ignored if the goal is NER. Removing twitter-specific or otherwise non-standard words showed a gradation of importance over the three tasks, with little importance for NER, moderate importance for parsing, and critical importance for speech synthesis. Capitalization correction had negligible impact on the parser or synthesizer, but was helpful for NER.

The importance of different edit types can be seen even at the most coarse level of examination. While normalization for speech synthesis is primarily dependent on removing unknown tokens, normalization that targets name entity recognition would be better served focusing on replacing non-standard to-

[5]Version 1.47.11, `http://espeak.sourceforge.net/`

kens with their standard forms. In contrast, parser-targeted normalization must attend to both of the tasks, as well as the task of restoring dropped tokens.

Despite the differences, there are a few common threads that appear in each evaluation. Most notably, the results suggest that the decision of most recent Twitter normalization work to focus on word replacement was not entirely without merit, as the high frequency of token replacements translated into high overall importance for all tasks. Similarly, the focus on slang was also somewhat reasonable, as failing to handle slang terms had a significant impact on parsing and speech synthesis, though it had little impact on entity recognition. Nonetheless, the results in Section 4 clearly suggest that handling these cases represent only a small fraction of the actions necessary to produce performance comparable to what would be seen on formal text.

Another similarity among all instances was the lack of importance of certain categories. For instance, punctuation addition was not important for any of the three tasks. While Zhang et al. had hypothesized that punctuation addition would be important for dependency parsing, the results given here suggest that the overall impact is minor. Similarly, contraction standardization was not shown to be important in any of the evaluations. Contraction normalization is more representative of how the normalization task was seen prior to the rise of social media normalization, as it represents a fairly minor normalizing action that might still be performed on formal text. Since contractions likely appear in a variety of forms in the data used to train NLP tools, it is unsurprising that these tools are comparatively robust to contraction differences than to cases that are less typically encountered.

6 Conclusion

In this work, we presented an in-depth look at the effects of the normalization of Twitter data. To do so, we introduced a taxonomy of normalization edits based on an examination of our Twitter dataset and inspiration from previous work. The taxonomy allowed for normalization edits to be examined systematically at different levels of granularity, and enabled an examination of the effects of not only token replacements, but the token additions and removals that recent work has suggested may have been unjustly ignored.

To understand the effects of each edit, we conducted ablation studies that examined results on three different downstream tasks: dependency parsing, named entity recognition, and text-to-speech synthesis. We found that while some normalization edits were universally important (or unimportant) for the production of accurate results, many differences persist. These results suggest that, for best results, how the normalization task is performed should not be agnostic of the downstream application. Further, our results support the suggestion that in order for downstream applications to produce accurate results, in most cases it is necessary to take a broad view of the normalization task the looks beyond simple word replacements.

Acknowledgments

The authors would like to thank Benny Kimelfeld for his comments on an early draft of this work. We also thank our anonymous reviewers for their constructive comments and feedback, and Stephanie Mcneish, Lacy Corlis, and Kaila Milos C. Factolerin for their assistance with annotation and evaluation.

References

AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for sms text normalization. In *ACL*, pages 33–40.

Tyler Baldwin and Joyce Chai. 2011. Beyond normalization: Pragmatics of word form in text messages. In *IJCNLP*, pages 1437–1441, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Richard Beaufort, Sophie Roekhaut, Louise-Amélie Cougnon, and Cédrick Fairon. 2010. A hybrid rule/model-based finite-state framework for normalizing sms messages. In *ACL*, pages 770–779.

Alan W. Black and Keiichi Tokuda. 2005. The blizzard challenge - 2005: evaluating corpus-based speech synthesis on common datasets. In *INTERSPEECH*, pages 77–80.

Monojit Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, and Anupam Basu. 2007. Investigation and modeling of the structure of texting language. *IJDAR*, 10(3-4):157–174.

Paul Cook and Suzanne Stevenson. 2009. An unsupervised model for text message normalization. In *CALC*, pages 71–78.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *NAACL-HLT*, pages 359–369, Atlanta, Georgia, June. Association for Computational Linguistics.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, pages 363–370, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Jennifer Foster, Özlem Çetinoglu, Joachim Wagner, Joseph Le Roux, Stephen Hogan, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011. hardtoparse: Pos tagging and parsing the twitterverse. volume WS-11-05 of *AAAI Workshops*. AAAI.

Bo Han, Paul Cook, and Timothy Baldwin. 2012. Automatically constructing a normalisation dictionary for microblogs. In *EMNLP-CoNLL*, pages 421–432.

Nobuhiro Kaji and Masaru Kitsuregawa. 2014. Accurate word segmentation and pos tagging for japanese microblogs: Corpus annotation and joint modeling with lexical normalization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 99–109, Doha, Qatar, October. Association for Computational Linguistics.

Catherine Kobus, François Yvon, and Géraldine Damnati. 2008. Normalizing SMS: are two metaphors better than one? In *COLING*, pages 441–448.

Chen Li and Yang Liu. 2014. Improving text normalization via unsupervised model and discriminative reranking. In *Proceedings of the ACL 2014 Student Research Workshop*, pages 86–93, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

Fei Liu, Fuliang Weng, Bingqing Wang, and Yang Liu. 2011a. Insertion, deletion, or substitution? normalizing text messages without pre-categorization nor supervision. In *ACL*, pages 71–76.

Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011b. Recognizing named entities in tweets. In *NAACL-HLT*, pages 359–367, Portland, Oregon, USA, June. Association for Computational Linguistics.

Fei Liu, Fuliang Weng, and Xiao Jiang. 2012. A broad-coverage normalization system for social media language. In *ACL*, pages 1035–1044.

Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC*, pages 449–454.

Deana Pennell and Yang Liu. 2010. Normalization of text messages for text-to-speech. In *ICASSP*, pages 4842–4845.

Deana Pennell and Yang Liu. 2011. A character-level machine translation approach for normalization of SMS abbreviations. In *IJCNLP*, pages 974–982.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in Tweets: An experimental study. In *EMNLP*, pages 1524–1534.

Cagil Sonmez and Arzucan Ozgur. 2014. A graph-based approach for contextual text normalization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 313–324, Doha, Qatar, October. Association for Computational Linguistics.

Richard Sproat, Alan W. Black, Stanley F. Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech & Language*, 15(3):287–333.

Pidong Wang and Hwee Tou Ng. 2013. A beam-search decoder for normalization of social media text with application to machine translation. In *NAACL-HLT*, pages 471–481, Atlanta, Georgia, June. Association for Computational Linguistics.

Yi Yang and Jacob Eisenstein. 2013. A log-linear model for unsupervised text normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 61–72, Seattle, Washington, USA, October. Association for Computational Linguistics.

Congle Zhang, Tyler Baldwin, Howard Ho, Benny Kimelfeld, and Yunyao Li. 2013. Adaptive parser-centric text normalization. In *ACL*, Sofia, Bulgaria, August. Association for Computational Linguistics.