# Syntactic Normalization of Twitter Messages

Max Kaufmann

***Abstract*—The use of computer mediated communication such as emailing, microblogs, Short Messaging System (SMS), and chat rooms has created corpora which contain incredibly noisy text. Tweets, messages sent by users on Twitter.com, are an especially noisy form of communication. Twitter.com contains billions of these tweets, but in their current state they contain so much noise that it is difficult to extract useful information. Tweets often contain highly irregular syntax and nonstandard use of English. This paper describes a novel system which normalizes these Twitter posts, converting them into a more standard form of English, so that standard machine translation (MT) and natural language processing (NLP) techniques can be more easily applied to them. In order to normalize Twitter tweets, we take a two step approach. We first preprocess tweets to remove as much noise as possible and then feed them into a machine translation model to convert them into standard English. Together, these two steps allow us to achieve improvement in BLEU scores comporable to the improvements achieved by SMS normalization**

## I. Introduction

TWITTER is a relatively new hybrid micro blogging/social networking website where users can post and read messages from a variety of electronic medium, such as Twitter's own website, text messages, or their computer desktop. Twitter is a popular medium for broadcasting news, staying in touch with friends, and sharing opinions. Since its initial founding in 2006, it has obtained over 100 million users [15]. Tweets, a term used to describe messages sent on Twitter, contain only 140 characters, 20 characters less than the 160 allowed by text messages. Twitter users are not even guaranteed to be able to use all of these for content. Twitter posts frequently included URLs, as well as markup syntax, which further decreases the amount of characters available for content. Because of these limits, users have created a novel syntax, very similar to SMS lingo, to communicate their messages with as much brevity as possible. While this brevity allows tweets to contain more information, it makes them harder to mine for information, due to its lack of standardization. Table 1 shows some examples of tweets.

TABLE I
Sample Tweets

| |
|---|
| Never say never.....dont let me goo dont let mee gooo dont let me gooooo.... |
| @user13431 when r u commin to Montreal |
| #bestfeeling is feeling like u mean the world to someone |
| My work buddy 'go smoke' like 3 times already |
| mai8mai RT @user1341 : Support Breast Cancer Awareness. Add A #twibbon To Your Avatar Now!! |
| I'm so #overyou Didn't even know it was possible!!! |

There are several issues that makes the normalization of tweets a difficult task. Tweets are written extremely colloquially, containing an unusually high amount of repetition, novel words, and interjections. A word may be written using a phonetic spelling (*nite* instead of *night*), or combined with other frequently used words into an acronym (*omg* instead of *oh my god*). Twitter users also have little regard for the proper use of capitalization and punctuation. Capitalization in a tweet may signal a proper noun or a sentence boundary, but it may also be used for something as arbitrary as emphasizing a certain segment. Punctuation may signal sentence boundaries, but it might also be used to create an emoticon. There are some deviations that are standard and systematic, but new variations can be created at any time, making the process of modeling the language extremely difficult. Additionally, Twitter users frequently use symbols to encode meta-content, such as who the tweet was directed to, or the topics to which it pertains. This meta-content sometimes is integrated into the syntax of the tweet, but there is no guarantee that it will be. In order to normalize these tweets, they will first be preprocessed to remove as much of the noise as possible, then fed into a machine translation model to convert them into standard English.

## II. Motivation

Due to Twitter's popularity, it has produced a massive amount of data. This data offers new and exciting opportunities, and there is much useful information that can be learned from meaningful analysis of this data. But the quality of the data is so poor that standard NLP tools are unable to process it. Tools such as Named Entity Recognizers have been shown to perform extremely poorly on tweets, most likely due to the high amount of noise present in tweets [5]. It has been shown that normalizing text messages allows standard MT techniques to work on them with little or no adaptation [3], and this paper posits that the same is true for tweets. If tweets can be converted to standard English, then the same should hold true for them. Another area in which data from Twitter can be used in is trend analysis. [4] claims that the unstructured nature of news articles, and the difficulty of NLP makes the problem of finding trends and topics in news articles a rather complex problem. These issues are magnified in tweets, which have all of the issues that normal news articles do, in addition to non standard orthography and extreme noisiness. Normalizing tweets would make work in this area, as well as any other area that involved analyzing tweets, much easier.

This is not to say that nobody has had success in mining data from tweets. Many studies have been able to draw conclusions from analyzing data on Twitter. Studies such as [2] have investigated how individuals use Twitter to communicate vital information in states of emergency. Papers such as [17] have shown that the informal communication that microblogs foster improves collaboration in the workplace. However, these studies concern the social effects of Twitter. Studies such as Puniyani et al. which attempt to preform an analysis focused on the content of the tweets admit that “Twitter contains highly non-standard orthography that poses challenges for early-stage text processing” [14].

## III. Previous work

While normalization of Twitter posts has never been attempted before, work has been done on noisy text normalization in the NLP field. However, tweets have several properties which makes normalizing them a substantially different problem than normalizing other forms of noisy text, such as emails or forum posts. First, they are very brief, containing only 140 characters. This means that it is much more difficult to use context as part of the disambiguation process. Tweets also have several novel syntactic elements which are especially challenging to disambiguate. Despite these differences, the process of tweet normalization is actually fairly close to the process of SMS normalization.

One area in which SMS normalization has been approached is to compare it to speech recognition. Text messages contain a significant number of tokens that are more indicative of its pronunciation, rather than its normal orthography.(e.g., *rite* instead of *right*)[9]. Speech recognition techniques are designed to decode phonetic representations into written words.[9] used techniques from automated speech recognition in order to normalize SMS. [9] claims that the dynamic nature of SMS is very difficult to capture with only a rule based MT system. They encoded the tokens in SMS messages into phonetic forms, and attempted to find the correct word with the most phonetic similarity.

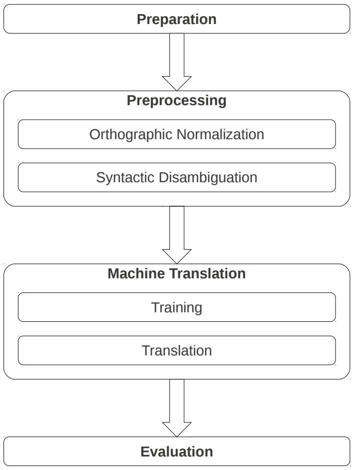
Another way to approach the problem is to look at tweets as though they are a different language, and attempt to use machine translation techniques to normalize them. This approach is fairly popular. Kobus et al [9] used this approach in addition to phonetic decoding. Others such as [8] and [3] have used supervised learning machine translation models to attempt to capture the most common SMS phrases and their English equivalents in a phrase table.

## IV. Normalization

Previous work such as [9] has suggested that combining multiple approaches in noisy text normalization creates the best results. Additionally, normalizing input text before inserting it into a MT system has been shown to improve the output quality. [13] modified their translations to harmonize word order, in order to improve the accuracy of their MT system. Preprocessing the tweets decreases the amount of noise present when they are being analyzed by the MT system, thus increasing the quality of the results. There are several issues, discussed in the following sections, which are easily dealt with by preprocessing, but would cause a great deal of confusion to a MT system.

The normalization model consists primarily of two parts, a normalization module and a statistical machine translation module. Below is a diagram.

Figure 1: Tweet Normalization Process



### A. Preparation

Before beginning the experiment, it was necessary to hand annotate tweets, so that there was a gold standard to evaluate the quality of the translation against . Approximately 1 million tweets were extracted from the the Edinburgh Twitter Corpus, a corpus containing 97 million Twitter posts[1]. These tweets were then filtered to remove tweets that were not in English. A tweet is not considered to be in English if it has under 40% English words. From these tweets, 1150 were randomly selected. These were hand translated by 10 annotators. The goal of this project is to remove as much noise as possible from tweets, so the annotators were instructed to remove any elements that were not absolutely necessary to form a grammatical English sentence. This included deleting elements such as smilies and extraneous punctuation, inserting subject pronouns, replacing acronyms, and correcting verb tenses. For example, the tweet *@user213 how are you?? I'm good :*” would have been translated as *How are you? I'm good*. This is a significantly different approach than is normally taken with SMS normalization. In SMS normalization corpora such as the ones created by [8], [6], many extraneous elements were kept in. The decision to remove them was motivated by the goal of this project. In order for tools such as named entity recognizers and semantic role labelers to work, they need their input to be as close to standard English as possible, and so it makes sense to remove these elements in tweets. Extra information would only serve to confuse these tools. In the normalization model, most of the extraneous elements are removed during preprocessing, so if a study were interested in using this extraneous information, it would be a fairly trivial task to leave these elements in.

### B. Orthographic Normalization

Although, Twitter messages are similar syntactically to SMS messages, they differ significantly in their type and quantity of orthographic errors. SMSs are created almost exclusively on cell phone, while approximately 90% of tweets come from the Web, IM, or custom applications, according to [11], all which use an automated spellchecker to suggest spelling corrections to the user. Because text messages are not spell checked prior to their submission, they are most likely going to have many more unintentional errors. The orthographic normalization model assumes that the main source of error in tweets will be from intentional spelling errors, since the spellchecker will have taken care of the majority of accidental errors. However, identifying spelling errors in Twitter messages is a difficult task. Twitter is frequently used as a medium to broadcast news [11] and therefore contain a large number of proper nouns that are not likely to be contained in a dictionary. Figuring out whether a word that is not in the dictionary is a misspelled or simply a novel word is not a trivial task. Because of this, the orthographic model is fairly conservative in spelling correction, to avoid misidentifying a novel word as a misspelled word. The approach taken simply identifies and corrects the most common intentional orthography errors.

One of the most common types of orthographical errors in Twitter posts is shortening words. Frequently used phrases are shortened into acronyms, and frequently used words are shortened by using phonetic spellings, or having characters removed. To help disambiguate these terms, a table of common SMS acronyms and short forms was created. This table was based on the work by [6]. [6]created a table containing common SMS errors and their English equivalents. This list was parsed to obtain a list of SMS acronyms that could be directly mapped to English words of phrases. The original table contained ambiguous mappings, such as translating *wt* as *what*, even though it is sometimes translated as *with*. If a statistical MT system were not part of the normalization approach, it might have been a good idea to leave the ambiguous mappings in, and just replace errors in the tweets with their most common correction. However, the statistical MT model is capable of disambiguating based on context, and can resolve ambiguous mappings. Therefore, all ambiguous sms terms which had multiple English mappings were stripped from the list by hand, leaving only items that could be unambiguously mapped to an English equivalent, such as *u, 2moro*, and *wut.*

One of the easiest spelling errors to make are off-by-one transpositions. For each misspelled word, all possible combinations that involve swapping two adjacent letters are tried, and if a correct match is found in the dictionary, the correct spelling is substituted.

Twitter messages frequently use repetition to convey emphasis. Written text lacks the tonality and variations which are used to convey emotions in spoken language, and so Twitter users are forced to creatively find ways to express emotion in their limited 140 characters. Similar to the way that people drag out words in spoken language to emphasize them, Twitter users frequently repeat characters in order to create emphasis. For example, a Twitter user wrote *OMG! I'm so guilty!!! Sprained biibii's leg! ARGHHHHHH!!!!!!* The repeated exclamation marks and extra letters on the token *argh* serve to emphasize the author's emotions. To correct for this, misspelled words that contain repeated sequential letters have these letters removed. If removing these letters creates a correct word, than the misspelled word is replaced in the text. Similarly, repeated punctuation is shortened to one punctuation mark, since the additional punctuation marks are not syntactically necessary.

### *C. Syntactic Disambiguation*

*1) @:* There are several elements in tweets that only sometimes have syntactic value. One of the elements is @username. Typing "@username" in a tweet is a processes commonly referred to as replying. However, the term "replying" is somewhat of a misnomer, as a user does not need to receive a message in order to "reply" with this syntax. The most general definition of this symbol is that it means the author of the post is telling a certain user that he thinks they would be interested in the content of the tweet. The most common use is to preface a tweet with @username (e.g., @Sammy wanna go to the park?). However, @username can appear at any point in the tweet. It can appear in the middle, if the author wants to address different sections of the tweet to multiple people (e.g., @sammy I'll be over tommorow @sally I'll fix it later). In these situations, @username has no syntactical value. However, that is not always the case. The @username is frequently incorporated into the sentence. For example, a user may write, *@sammy is my best friend*! or *Im at the park with @sammy and we're having a great time*. In these situations, the @username performs a syntactic role in the sentence, and its removal would be grammatically improper.

Another feature of the @username syntax is that it can be used to broadcast information to all of a users followers. Twitter users frequently send tweets they find interesting to all of their followers with the syntax "RT @username:", where username is the username of the original author, of the followed by the original message. While this appears similar to @username, it is syntactically different. RT @username almost always has no syntactic value, and can be removed while maintaining proper syntax.

Analysis of these tweets has revealed that there are certain linguistic properties that can be analyzed to remove the majority of @username when it is appropriate. By tagging the text that is being translated with part of speech (PoS) tags, it becomes apparent that when the @username needs to be kept in the sentence for syntactical reasons it is preceded or followed by certain parts of speech. If the @username is at the beginning of the tweet, then only the subsequent terms can be used in this analysis. If the @username is followed by a word that is either a coordinating conjunction, subordinating conjunction, preposition,or a verb it is almost always necessary to keep the @username in the tweet. If it is not the first word, then the part of speech of words on both sides can be used to help disambiguate @username. In these situations, the preprocessor checks for the above conditions, but it also checks to see if the part of speech of the preceding word belongs to the previous list.

*2) #:* Another element which may or may not have syntactic value is the #. The most common syntax of # (read as hash or hash symbol) is #topic. The word following the # is generally the topic to which the tweet pertains. If a user was tweeting about the government stimulus bill, they might insert *#stimulus* into their tweet. This process is called tagging, and a #topic is commonly known as a tag. Twitter uses these tags to classify posts by common topic. Like @, the # may or may not have syntactic value. It is most commonly inserted at the end of a tweet without any syntactic value (e.g., I just got the new Droid phone #droid), but if the topic of the tweet is contained within the tweet, users frequently append a hash to the topic, in order to stay within the 140 character limit and avoid repetition (e.g., I just got the new #droid phone). Additionally, in tweets that are about the user's mood instead of a certain topic, it is common to use the # for emphasis (e.g., At work thinking abt how I have to leave tonight and come Right back in the am. #Argh And I have to train somebody 2day).

Unfortunately, PoS tags alone cannot be used to decide whether a hashtag has syntactic value in the sentence. @username is always a noun, whereas a hashtag can be any type of word. Observation of the annotated tweets shows that humans almost always thought that terms with a hashtag located in the middle of a sentence were important to the syntax of the sentence, and should not be deleted. However, hashtags at the beginning and end of tweets are much more difficult to disambiguate. This is an unfortunate problem, since the beginning and end of tweets are where hashtags are most commonly found. To disambiguate these hashtags the following heuristics were used: If there are two or more sequential hashtags, it is likely that they are topics, and have no syntactic value, and so they can be removed. Additionally, if a hashtag is preceded by a terminal punctuation mark, we can assume that they are standalone topics, and play no role in the syntax of the tweet, and can also be removed. If the hashtag is preceded by a conjunction, preposition, or transitive verb, then we can assume that the hashtag is syntactically linked to the previous term and needs to remain in order to preserve the syntax of the tweet. If none of these conditions were met, then the hashtag was removed.

### *D. Statistical Machine Translation*

After the preprocessing is done, the tweets are ready to be fed into the statistical machine translation system. The tool that was used to build this system is Moses. Moses is a statistical machine translation package which can produce high quality translations from one language into another[10]. At its core, translation simply consists of finding phrases in one language that correspond to phrases in another language. While the tasks of tweet normalization is not translation, it does consist of converting one set of phrases into another set, which makes Moses an extremely valuable tool.

*1) Training:* According to the Moses website [1], there are 9 steps involved in creating a Moses model

1) Prepare data
2) Run GIZA++
3) Align words
4) Get lexical translation table
5) Extract phrases
6) Score phrases
7) Build lexicalized reordering model
8) Build generation models
9) Create configuration file

GIZA++ is a tool which attempts to align the words from one corpus which their equivilant, or equivilants in another. When translating from one language to another, this is a difficult task, since one word in one language may correspond to several words in another. However, when translating from tweets to normal English, this is a fairly trivial task, since most of the words have a one to one mapping. Step 3 simply uses heuristics to increase the accuracy of the word alignmetns suggested by Giza. The result of all this is step 4, a lexical translation table. This table simply gives the probability for w(e|t), where e is an English word, and t is a word in Twitter English. Based on this lexical translation table, and the alignments created by GIZA++, Step 6 can created a phrase translation table, which is similar to the lexical translation table, except that it contains the probabilities of phrases in a tweet being translated as a particular English phrase. Steps 7 and 8 refer to steps which are relevant to the change in word order that comes from translation, and reverse translations. These steps are not relevant to the processes being discussed in this paper and so will not be discussed.

Before Moses can be used to produce translation, it must be trained on a data set, so that it can learn the rules that govern the translation. Training Moses requires a corpus in the target language, from which an *n-gram* language model (LM) is built. In this experiment, the LM was built from the Open American National Corpus (OANC)[7], a corpus of 15 million words from a variety of contexts. Moses also requires a set of parallel corpora, one in the source and one in the target language. This posed a significant problem, since there are currently no annotated tweets which could be used as corpora. To resolve this issue, a set of parallel SMS corpora was used. These corpora were created by [8], who generously made them available for use in this experiment. The corpora consist of approximately 18,000 text messages, gathered from various sources. They were annotated by the two authors of [8], who did not use inter-annotator agreement to validate their results. While this would be an issue if this were a translation problem, normalizing text messages and Twitter posts is a much more akin to correcting grammar than translating text from one language to another, and so inter-annotator agreement is not necessary.

*2) Translation:* Since the task at hand is not truly translation from one language to another, several of Moses' default settings have to be tweaked in order create a high quality translation. The first is the distortion limit. Translating from one language to another often requires heavy reordering of the words. By default, Moses will allow the reordering of phrases up to 7 words long. This feature would be useful in a situation involving translating into a target language that had a word order very dissimilar to the source language. However, in this context, Twitter English lines up at almost a one-to-one ratio with normal English. Several settings were tested, and

[1] http://www.statmt.org/moses/?n=FactoredTraining.HomePage

the results indicated that the distortion limit made very little difference, similar to the findings in [8]. They used Moses for the normalization of text messages, and found that most of the phrases learned by the system only involved a one-to-one mapping. So even when the translation model was allowed to alter the word order, it chose not too.

Each element of the translation module of Moses is weighted. The weights of the translation model tell Moses how much emphasis should be placed on certain factors, such as the *n-gram* ordering derived from the language module, in the translation process. By adjusting the weight of the language model (LM), it was found that the BLEU scores could be increased by approximately .4 if the LM had a weight of .3, instead of its default weight of 1. This means that the *n-grams* generated from the OANC were not considered very important when translating. This makes sense, given that there is very little overlap between the domain of the OANC and the domain of this project.

An additional feature in Moses is the recaser. The recaser was originally designed to correctly case text if it had been translated in lowercase. In previous experiments on SMS normalization, the issue of case was ignored. However, it is very important in tweet normalization, because tweets contain so many proper nouns. In this experiment, the recaser was trained on the LM built from the OANC. When the tweets were originally translated into English, all previously seen tokens were lowercased. This was because the original capitalization of a tweet is not a reliable indicator of the true capitalization. Twitter users frequently use capitalization as emphasis, by either capitalizing the entirety of a word, or the first letter of a series of words. Unknown tokens were left in their original case, because they were most likely to be proper nouns or acronyms.

The recaser uses techniques similar to those outlined in [12]. This involves building a trigram language model, and using that to compute the probabilities of the most likely case [12]. For example, *new* is almost always lowercased, but when it is followed by *York*, it is almost always capitalized. This technique seems to be very successful in correctly casing the text, except when commonly seen words appear in a novel sequence that requires them to be capitalized. For example, the tweet *@user1941 The film I really want to see at the mo is Men Who Stare at Goats* was translated as “The film i really want to see at the moment is men who stare at goats”, because the tokens *men, who, stare,* and *goats* were almost exclusively lowercased in the OANC. However, in situations where Twitter users forgot to capitalize commonly used proper nouns, the system preformed very well. The tweet *HOME ALONE RT @user3413 : I'm craving for christmas movies!! any suggestion??* was translated as *Home alone I am craving for Christmas movies! Any suggestion?*, successfully lowercasing the capitalized text at the start, and uppercasing the proper noun “Christmas.”

## V. Evaluation

The goodness of a translation is judged is using the BLEU score. The BLEU score is a tool designed for evaluating the accuracy of translations from one language to another. A BLEU score requires a gold standard, which contains the translations as done by human. This file is compared against a machine translated version, and is then assigned a score between 0 and 1. A score of 1 would indicate that the machine translated version is exactly the same as the human translated version, while 0 means that the two versions are very different. The language of a tweet is so different from the normalized result that this tool should provide an accurate indication of how well the translation worked. Below are the BLEU scores of the translation before and after normalization. NIST, an alternate MT scoring metric, scores are included in the table, so that future papers who choose to evaluate their work with NIST will have a baseline to compare their results against.

TABLE II
Evaluation of results

| | BLEU scores | NIST scores |
|---|---|---|
| Before Normalization | 0.6799 | 10.5693 |
| After Normalization | 0.7985 | 11.7095 |

The results indicate that the normalization process had a significant effect on BLEU scores, increasing them by 18%. Since Twitter normalization has never been undertaken before, there are no results against which these scores can be compared. The closest available data is data regarding SMS normalization. Below is a table of several papers on SMS normalization and the BLEU scores they achieved.

TABLE III
SMS Normalization scores

| | Kobus et al. | Karthik and Krawczyk | Chourhury et al. |
|---|---|---|---|
| Before Normalization | n/a | .54 | .57 |
| After Normalization | approx. .8 | .8611 | .8070 |

These data shows that tweets appear to have much higher initial BLEU scores. This supports the theory that tweets do not have as many orthographical errors as SMS messages. Acronym usage and misspelling are much more common in SMS than in tweets. This is demonstrated in [8], where they were able to boost the BLEU score of a test set from .562 to .794 by simply performing a search and replace for common texting acronyms. In comparison, the preprocessing module of this normalization system, which included a very similar search and replace, only boosted the initial score of .6799 to .7324. This seems to suggest that while SMS may contain more errors to begin with, they contain a far more regular pattern of errors. The difficult part of tweet normalization is removing noisy elements which prevent the tweet from being mapped to an English syntax. So, while SMS and tweet normalization are attempting to achieve similar goals, the problems they attempt to solve are actually fairly different.

### A. Sample Translations.

Below are several translations generated by the normalization system. The <URL> tag signifies that the original tweet contained a URL in that location.

***Original:*** @user3419 nay lol y u say dat?&wat u doing 2day?

***Post-normalization:*** No, why did you say that? What you doing today?

***Original:*** 1001 colors: Contemporary art from Iran <URL> #Iran #culture #Art

***Post-normalization:*** 1001 colors: contemporary art from Iran <URL>.

***Original:*** it's soo quiet, it's like I'm goin die

***Post-normalization:*** It is so quiet, it is like i am going to die.

***Original:*** #worstfeeling buyin a fresh laptop..then ur screen blowz out :((

***Post-normalization:*** worst feeling is buying a fresh laptop.. then your screen blowz out.

***Original:***This is superb Grape+apple splash with manggo juice, super!

***Post-normalization:*** This is superb grape + Apple splash with mango juice, Super!

***Original:*** @user31903 u n ur fam can n if u interested ill b n touch w u bout it

***Post-normalization:*** You and your family can and if you interested Ill be and touch with you about it.

***Original:*** RT: @user4191 BEAUTIFUL CREATURES has a new #website designed by @user4192!

***Post-normalization:*** Beautiful creatures has a new website designed by @user4192!

From these tweets, it is apparent that the syntactic disambiguation module of the preprocessor is able to discern whether the syntactically ambiguous elements in the tweet are necessary or not. The model successfully removes it when it has no direct mapping to English syntax, and keeps it when it is necessary. The output of the normalization system produces much more readable results, removing extraneous noise that doesn't map to English syntax.

However, there are some issues with the normalization system. There are some orthographic errors that are not caught. The orthographic normalization system does not deal with phonetic substitutions very well, such as *blowz* instead of *blows*. Dealing with these requires a very sophisticated model, such as the one created by [6]. We feel that these errors are rare enough that the additional computational complexity required by these models is not justified in this system. However, future work attempting to improve the quality of the results could implement a system like this.

## VI. Possible improvements

### A. Metrics

One possible area of improvement involves finding a better metric to evaluate noisy text normalization. While previous researchers studying SMS normalization have chosen to evaluate their results with the BLEU metric, it might not be the best choice. The BLEU scoring metric was designed for evaluating translations from one language to another, not for evaluating the results of noisy text normalization. Because of this, a better BLEU score does not necessarily mean a better translation. For example, the subjectivity of the human annotators could cause substantial variation in BLEU scores. In papers such as [8] their corpora was only annotated by two people. The fact that there were 10 annotators could have lead to inconsistencies in the scoring data. For example, although annotators were instructed to expand contractions, some annotators chose to translate *Im* as *I'm*, instead of *I am*. BLEU scores are obtained by comparing the similarities between *n-grams* of the hypothesized translation and gold standards, so errors such as this could have detrimental effects on the score, despite the fact that "I'm" and "I am" are grammatically equivalent.

Even if we ignore the issue of the applicability of BLEU as an evaluator itself, there are still several problems with the BLEU metric itself. The relationship between BLEU scores and human judgment is questionable. Papers such as [16] have suggested that an increase in BLEU score may not correlate with an increase in translation quality. In fact, on a test of several machine translation systems, the correlation between human and BLEU scores was found to be as low as .38 in some cases. One example where the BLEU score performed poorly was on the tweet *@user12493 I'm following u now should I hold on tight?*. The translation generated by the normalization system was *I'm following you now, should I hold on tight*". This seems like a perfectly acceptable translation. However, the human annotator translated the tweet as *I'm following you. Now, should I hold on tight?*. BLEU scores this translation at .43. However, both translations are acceptable.

### B. Corpora

Besides improving the scoring metric, there are several ways in which the translation process can be improved. The easiest improvement would be to use tweets as training data, instead of text messages. Constructing an annotated twitter corpora would be a difficult and time consuming task, but would allow the MT model to do much of the work done in the preprocessor, such as syntactic disambiguation of @username or #tag. Providing more detailed data would also improve the quality of the results. Moses has the ability to incorporate additional lexical information such as PoS tags into its translation model. Including this information would allow Moses to create more sophisticated rules governing the translation from tweets to English.

A better language model would also improve the quality of the translation results. The current corpus used to build the language model, the OANC is not especially representative of the structure of tweets, as evidenced by the fact that decreasing its weight in the translation process from 1 to .3 resulted in an better results. Perhaps a language model built from text messages, or tweets, would be better. This study attempted to build a LM from the SMS corpora provided by [8], but it did not improve the quality of the results. However, this is probably due to the fact that the OANC contains magnitudes more data than the SMS corpus. If an SMS corpus of sufficient size could be obtained, it would probably create a much more applicable language model.

### C. Utilizing Additional Properties

There are additional ways in which semantic information of tweets could be used to aid in the normalization process that fall outside the scope of this paper. While there are many acronymns that are standard across Twitter, there is no official standard language. Because of this, it is difficult to draw conclusions about the nature of the language used on Twitter by looking at a large set of tweets. However, it might be possible to use Moses to create localized translation models by looking at smaller subsets of tweets. For example, tweets from users who reside in a particular nation might have their own set of slang. Tweets that are obtained from Twitter include information about the users location, as well as their country of orgin (if they have elected to include that information). This information could be leveraged to create a localized corpora which can more accurately translate slang from a certain region.

There are other factors besides semantic information that could be used to improve the quality of the translations. Many papers such as [6] have used phonetic systems to do orthogrpahic normalization, and have achieved a fair degree of success. This approach could be combined with the ones mentioned in this paper fairly easily. Additionally, heuristics could be used to combine the possible outputs of a phonetic system with the results of this system to decrease overall error. For example, a phonetic system would realize that *rite* is phonetic approximation of the word *right*. This would help disambiguate between other possible spelling suggestions, such as *write*. In turn, the semantic properties of the tweet could be used to decide if the usage of *rite* is a mispelling or not.

## VII. Conclusion

In this paper, it was shown that combining statistical machine translation software with a preprocessor, it is possible to remove the majority of noise from a tweet, and increase its readability significantly. The benefits of this study are a novel system which can successfully map a tweet to a syntactically correct English sentence. It seems that the results of this study are sufficiently accurate enough to allow tweets to be mined for data. Additionally, now that the tweets conform to normal English syntax, NLP tools such as part of speech taggers, document summarizers, named entity recognizers, or semantic role labelers should achieve much better performance.

The value of the work in this paper is in its applicability to other procedures. One of the applications that should so significant performance when combined with this tool is a Twitter post summarizer. David Inoyue is currently working on a program that produces multiple sentence summaraizes about Twitter posts on one topic. The resulting summarizies are made up of the tweets in that topic. Since the summaries are made of tweets, normalizing them should make them significantly more readable. David is currently in the process of measuring the success of his summarizer, and when he is done we will include the results that normalization had on this process in this paper.

## VIII. Acknowledgments

The author of this paper would like to thank Karthik Raghunathan and Stefan Krawczyk for making their SMS corpora available.

## References

[1] *The Edinburgh Twitter Corpus*, Los Angeles, California, June 2010. Computational Linguistics in a World of Social Media.

[2] *The Nays Have It: Exploring Effects of Sentiment in Collaborative Knowledge Sharing*, Los Angeles, California, June 2010. Computational Linguistics in a World of Social Media.

[3] AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. A phrase-based statistical model for sms text normalization. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 33–40, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[4] Daniel Billsus and Michael J. Pazzani. A personal news agent that talks, learns and explains. In *AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*, pages 268–275, New York, NY, USA, 1999. ACM.

[5] James Martin Brian Locke. Named entity recognition: Adapting to microblogging. Master's thesis, University of Colorado, 2009.

[6] Monojit Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, and Anupam Basu. Investigation and modeling of the structure of texting language. *Int. J. Doc. Anal. Recognit.*, 10(3):157–174, 2007.

[7] Nancy Ide and Catherine Macleod. The american national corpus: A standardized resource for american english. In *Proceedings of Corpus Linguistics 2001*, pages 831–836, 2001.

[8] Stefan Krawczyk Karthik Raghunathan. Investigating sms text normalization using statistical machine translation. Stanford University, Stanford, CA, 2009.

[9] Catherine Kobus, François Yvon, and Géraldine Damnati. Normalizing sms: are two metaphors better than one? In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, pages 441–448, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

[10] Phillip Koehn and Hieu Hoang. Moses: Open source toolkit for statistical machine translation. Technical report, Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prauge, Czech Republic, June 2007.

[11] Balachander Krishnamurthy, Phillipa Gill, and Martin Arlitt. A few chirps about twitter. In *WOSP '08: Proceedings of the first workshop on Online social networks*, pages 19–24, New York, NY, USA, 2008. ACM.

[12] Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. truecasing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 152–159, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[13] Sonja Nieben, Hermann Ney, and Lehrstuhl Fur Informatik Vi. Morpho-syntactic analysis for reordering in statistical machine translation, 2001.

[14] Kriti Puniyani, Jacob Eisenstein, Shay Cohen, and Eric P. Xing. Social links from latent topics in microblogs. In *Conference on Social Media*, page 31, June 2010.

[15] Reuters. Twitter snags over 100 million users, eyes money-making. April 2010.

[16] Ying Zhang, Stephan Vogel, and Alex Waibel. Interpreting bleu/nist scores: How much improvement do we need to have a better system. In *In Proceedings of Proceedings of Language Resources and Evaluation (LREC-2004*, pages 2051–2054, 2004.

[17] Dejin Zhao and Mary Beth Rosson. How and why people twitter: the role that micro-blogging plays in informal communication at work. In *GROUP '09: Proceedings of the ACM 2009 international conference on Supporting group work*, pages 243–252, New York, NY, USA, 2009. ACM.