TEXT NORMALIZATION USING LEXICAL AND CONTEXTUAL FEATURES

by

Çağıl Uluşahin Sönmez

B.S., Computer Science, Istanbul Bilgi University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2014

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my master thesis supervisor Assist. Prof. Arzucan Özgür. I have learned many things since I became her student. Her positiveness and encouragement made me complete all those tasks that once seemed impossible. Without her support and guidance I would not be able to complete this thesis successfully.

I would also like to thank my committee members Prof. Tunga Güngör and Assist. Prof. Gülşen Cebiroğlu Eryiğit for kindly accepting to be in my thesis committee and for their contribution and support to this work. I also thank my academic advisor Prof. Fatih Alagöz for his valuable guidance during my master education.

I would like to express my appreciation to Bilgi University Computer Science Department faculty members. Your touch in my path of life and education is priceless, thank you for being a part of my life for such a long time. At each step of my life I always remember and appreciate each and everyone of you, your guidance and support.

I am also thankful for the help and support I have received from my fellow graduate students Arda Çelebi, Haşim Sak, Ahmet Yıldırım and Onur Güngör.

My family is always believed in me and supported me. I am deeply grateful to my mother, father and brother for supporting me and my decisions in each way possible. I would like to thank my mother in law and father in law for their support and love. I consider myself lucky for having such a wonderful family.

At the end I would like express appreciation to my beloved husband Ahter Sönmez who spent sleepless nights with and was always my support in every possible moments. Your advice on both research as well as on my career have been priceless. Words fail to express my gratitude to you.

ABSTRACT

TEXT NORMALIZATION USING LEXICAL AND CONTEXTUAL FEATURES

The informal nature of social media text, renders it very difficult to be automatically processed by natural language processing tools. Text normalization, which corresponds to restoring the noisy words to their canonical forms, provides a solution to this challenge. We introduce an unsupervised text normalization approach that utilizes not only lexical, but also contextual and grammatical features of social text. The contextual and grammatical features are extracted from a word association graph built by using a large unlabeled social media text corpus. The graph encodes the relative positions of the words with respect to each other, as well as their part-of-speech tags. The lexical features are obtained by using the longest common subsequence ratio and edit distance measures to encode the surface similarity among words, and the double metaphone algorithm to represent the phonetic similarity. Unlike most of the recent approaches that are based on generating normalization dictionaries, the proposed approach performs normalization by considering the context of the noisy words in the input text. Our results show that it achieves state-of-the-art F-score performance on a standard data set. In addition, the system can be tuned to achieve very high precision without sacrificing much from recall.

ÖZET

KELİME VE BAĞLAM BİLGİSİ TEMELLİ METİN NORMALİZASYONU

Sosyal medya metinlerinde kullanılan dilin bozukluğu bu metinleri doğal dil işleme araçları ile otomatik olarak işlemeyi çok zorlaştırmakta. Bu bozuk metinleri düzeltip kitap biçimlerine dönüştürme bir diğer deyişle metin normalizasyonu, bu soruna bir çözüm ortaya koymaktadır. Bu çalışmada, sosyal metinlerin sözcüksel ve içeriksel özelliklerinin yanısıra dibilgisi özelliklerinden de faydalanılan gözetimsiz bir metin normalizasyonu yaklaşımı sunuyoruz. İçeriksel ve dilbilgisel özellikler, büyük ve etiketlenmemiş bir sosyal medya derlemi kullanarak oluşturduğumuz kelime ilişkilendirme çizgesi yardımı ile hesaplanıyor. Bu çizge, kelimelerin metin içerisinde birbirleriyle olan konum ilişkilerini ve cümle öğe bilgilerini (part-of-speech) içermektedir. Sözcüksel özellikleri bulmada kelimelerin en uzun ortak altdizileri ve birbirine dönüşme uzaklıkları gibi yazım benzerlikleri yanısıra çift metafon (double metaphone) gibi ses bilimsel benzerlikleri göz önünde bulunduran yöntemlerden faydalanıldı. Yakın zamanda sıkça kullanılan sözlük bazlı çalışmaların aksine, önerdiğimiz yaklaşım metin normalizasyonunu düzeltilecek metnin içeriğini göz önünde bulundurarak uygulamaktadır. Standart veri kümesi üzerinde literatürdeki sonuçlardan daha yüksek sonuçlara ulaşan sistemimiz farklı parametreler kullanılarak kapsama (recall) degerinden ödün vermeden çok daha yüksek kesinlik (precision) değerlerine ulaşabilmektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\mathrm{CL}(o_i)$	Candidate List of o_i
c_k	k^{th} Candidate Word/Node for the OOV Token in the Given Input Text
$\mathrm{EL}(o_i)$	Edge List of o_i
$\mathrm{NL}(o_i)$	Neighbor List of o_i
n_j	j^{th} Neighbour of the OOV Token in the Given Input Text
o_i	i^{th} OOV Token in the Given Input Text
$t_{distance}$	Word Distance Threshold for Contextual Association
t_{edit}	Edit Distance Threshold
$t_{frequency}$	Word Frequency Threshold
T_i	Tag of a Word
$t_{phonetic}$	Phonetic Edit Distance Threshold
β	Contextual Similarity Minor Score Parameter
λ	Lexical Similarity Minor Score Parameter

LIST OF ACRONYMS/ABBREVIATIONS

A	Adjective
C	Punctuation
CMU	Carnegie Mellon University
CWA-Graph	Contextual Word Association Graph
D	Determiner
ED	Edit Distance
G	Miscellaneous words
IV	In Vocabulary
L	Nominal+verbal
LCS	Longest Common Subsequence
LCSR	Longest Common Subsequence Ratio
MT	Machine Translation
N	Noun
NLP	Natural Language Processing
OOV	Out of Vocabulary
P	Preposition
POS	Part of Speech Tag
RT	Retweet
SMS	Short Messaging Service
STT	Speech To Text
URL	Uniform Resource Locator
V	Verb

1. INTRODUCTION

Within the last decade, the common belief among Internet users that social text has (or should have) its own lexical and grammatical features has naturally given birth to an Internet language and jargon; which has been steadily growing and evolving ever since [4,5]. This behavioral preference phenomenon brings another challenge of its own. Not only is the Internet jargon itself growing and evolving in an exponential pace, but also since the beginning of the World Wide Web, the Internet has its own slang. *lol* meaning *laughing out loudly*, *xoxo* meaning *kissing*, *4u* meaning *for you* are among the most commonly used examples of this slang. In addition, these specific forms of informal expressions in social text usually take many different lexical forms when generated by each individual, even though the intended contextual meaning might be the same [5]. In other words, with each different individual the same content is being expressed (written) in different ways. Due to this unpredicted variety of such expressions, it would be appropriate to call this divergency "noise" in social text.

The scope of the problem does not end there. In addition, within the last few years, by the increasing use of mobile devices, social text has now been preferred to be transcribed by using Speech-to-Text (STT) tools. This text input preference is getting trendier and being used more frequently. The insufficient accuracy of such STT tools brings considerable amount of "additional noise" to social text. Tools such as spell checkers and slang dictionaries have been shown to be insufficient to cope with this challenge long time ago [6].

Lastly, when we also consider the usual scarcity of attention when people post messages on social media platforms, the problem of analyzing social text actually goes beyond the reach of human cognitive capacity. The mass usage of such social media platforms makes it impossible to derive analysis results in a limited time scope when processed manually. In addition, most automatic Natural Language Processing (NLP) tools such as named entity recognizers and dependency parsers generally perform poorly on social media text [7].

Text normalization is a preprocessing step to restore noisy words in text to their original (canonical) forms [8] to make use in NLP applications or more broadly to understand the digitized text better. For example, *talk 2 u later* can be normalized as *talk to you later* or similarly *enormoooos, enrmss* and *enourmos* can be normalized as *enormous*. You can find more examples of normalized text in Table 1.1. These noisy tokens are referred as Out of Vocabulary (OOV) words. The normalization task restores the OOV words to their In Vocabulary (IV) forms. Table 1.2 shows sample OOV words encountered in social media text and their corresponding IV forms.

Table 1.1. Sample tweets and their normalized forms.

Its a beautiful nite, lukin for smth fun to do, I think I wanna be w ma frnds.	It's a beautiful night, looking for something fun to do, I think I want to be with my friends.
Dnt always follow da crowd, stand 4 wat u blv in.	Don't always follow the crowd, stand for what you beleive in.
@Cloudy me tht go be sad wen the hangover hold me tmr!	@Cloudy me that going to be sad when the hangover hold me tomorrow!
I srsly need some legend of korra raight nao #linplz	I seriously need some legend of korra right now #linplz
Wat was tht for u lil shit, dnt u draw on my enlgand	What was that for you little shit, don't you draw on my England
Work f a cos, not for applause. Live life to exprss, not to imprss :)	Work for a cause, not for applause. Live life to express, not to impress :)
*Hav guts to say wat u desire.. Dnt beat behind da bush!! And 1 mre thng no mre say **y** r people's man!!*	Have guts to say what you desire.. Don't beat behind the bush!! And one more thing no more say **you** are people's man!!
*There r sm songs u don't want 2 listen 2 yl walking cos when u start dancing ppl won't knw **y**.*	There are some songs you don't want to listen to while walking because when you start dancing people won't know **why**.

Table 1.2. Sample noisy tokens and their normalized forms.

Ill-formed word	**Normalization**
ppl	people
tmr	tomorrow
havent	haven't
soooo	so
raight	right
raight	alright
cos	because
cos	cause
r	are
n	and
mor	more
doin	doing
finge	finger
tnks	thanks
makeing	making
friiied	fried

Every OOV word should not be considered for normalization. Social text is continuously evolving with new words and named entities that are not in the vocabularies of the systems [9]. For example *iPhone, WikiLeaks or tokenizing* have not taken their places in dictionaries yet, so they are OOV words, but they should not be normalized to any other canonical word. In addition, an OOV word can sometimes lexically fit an IV word (Ex: *tanks* is both an IV word and an OOV word with the canonical form *thanks*).

The OOV tokens that should be considered for normalization are referred to as ill-formed words. Ill-formed words can be normalized to different canonical words depending on the context of the text. For example, if we look at last two examples in Table 1.1, we see that "y" is normalized in the first as "why" and as "you" in the latter. Another example would be "cos", it has two common canonical forms "cause" and "because".

In [4] Choudhury *et al.* propose that OOV words observed in noisy text can be classified into two groups, unintentional and intentional errors. The unintentional errors are caused by (i) pressing of the wrong key, (ii) pressing of a key more than the desired number of times, (iii) deletion of a character or (iv) inadequate knowledge of spelling. As for the intentional errors, they can be categorized into four categories: character deletion("tlk" for "talk", "msg" for "message", "tomoro" for "tomorrow", "mob" for "mobile"), phonetic substitution ("nite" for "night", "bk" for "back", "u" for "you", "m8" for "mate"), abbreviations ("btw" for "by the way", "kgp" for "Kharagpur") and non-standard usage ("wanna" for "want to", "betta" for "better", "sumfin" for "something", "b/c" for "because").

In this thesis, we propose a graph based text normalization method that utilizes both contextual and grammatical features of social text. The contextual information of words is modeled by a word association graph that is created from a large social media text corpus. The graph represents the relative positions of the words in the social media text messages and their Part-of-Speech (POS) tags. The lexical similarity features among the words are modeled using the longest common subsequence ratio and edit

distance that encode the surface similarity and the double metaphone algorithm that encodes the phonetic similarity. The proposed approach is unsupervised, which is an important advantage over supervised systems, given the continuously evolving language in the social media domain. The same OOV word may have different appropriate normalizations depending on the context of the input text message. Recently proposed dictionary-based text normalization systems perform dictionary look-up and always normalize the same OOV word to the same IV word regardless of the context of the input text [8, 9]. On the other hand, the proposed approach does not only make use of the general context information in a large corpus of social media text, but it also makes use of the context of the OOV word in the input text message. Thus, an OOV word can be normalized to different IV words depending on the context of the input text. Another strength of the proposed system is that it achieves the state-of-the art precision scores, without sacrificing from recall.

2. RELATED WORK

Early work on text normalization mostly made use of the noisy channel model. The first work that had a significant performance improvement over the previous research was by Brill and Moore, 2000 [10]. They proposed a novel noisy channel model for spell checking based on string to string edits. Their model depended on probabilistic modeling of sub-string transformations. They ran their experiments first by using the error model in isolation assuming that each word in the dictionary has uniform probability. Their results showed that the longer the word the better performs their character level noisy model. When they used a trigram language model that they built by using the Brown corpus instead of the uniform distribution, their results improved.

Toutanova *et al.*, 2002 improved this approach by extending the error model with phonetic similarities over words [11]. Their approach is based on learning rules to predict the pronunciation of a single letter in the word depending on the neighbouring letters in the word. They used a trigram phone sequence language model and a fourgram vowel sequence language model to re-rank the top n results. In addition, they distinguish between the middle of the word versus the start and end of the word and interpolate their model with the letter based model of Brill and Moore, 2000 [10]. Their extended and combined model substantially reduced the error rate of Brill and Moore's model, and performed best at top 3 results.

Choudhury *et al.*, 2007 developed a supervised Hidden Markov Model based approach for normalizing Short Message Service (SMS) texts [4]. They proposed a word for word decoding approach and used a dictionary based method to normalize commonly used abbreviations and non-standart usage (e.g. "how are" to "howz" or "are not" to "aint"). Cook and Stevenson, 2009 have extended this model by introducing an unsupervised noisy channel model [12]. Rather than using one generic model for all word formations as in Choudhury *et al.*, 2007, they used a mixture model in which each different word formation type was modeled explicitly.

The down side of these methods were: (i) they did not consider contextual features and (ii) each of them assumed that tokens have unique normalizations. However, that is not the case for the normalization task. The OOV tokens are ambiguous and without contextual information it is not possible to build models that can disambiguate transformations correctly.

Aw *et al.*, 2006 proposed a phrase-based statistical machine translation (MT) model for the text normalization task [13]. They defined the problem as translating the SMS language to the English language and based their model on two submodels: a word based language model and a phrase based lexical mapping model (channel model). Their phrase based model is an extended noisy channel model which does "many word" to "many word" mappings such as "ysnite" $\rightarrow$ "yesterday night". Their system also benefits from the input context and they argue that the strenght of their model is in its ability to disambiguate mapping as in "2" to "two" or "to", and "w" to "with" or "who". Making use of the whole conversation, this is the closest approach to ours in the sense of utilizing contextual sensitivity and coverage.

Pennell and Liu, 2011 [14] on the other hand, proposed a character level MT system, that is robust to new abbreviations. Their system has two phases. In the first phase, a character level trained MT model is used to recognize common abbreviations and to produce word hypotheses (making use of CMU lexicon). In the second phase, a trigram language model is used to choose a hypothesis that fits into the input context. They also used a reordering model and a word length penalty while scoring the assigned translation.

The models described above are supervised models a drawback of which is that they require annotated data. Annotated training data is not readily available and is difficult to create especially for the rapidly evolving social media text [15].

More recent approaches handled the text normalization task by building normalization lexicons. Han *et al.*, 2011 developed a two phased model, where they only consider the ill-formed OOV words for normalization [8]. First a confusion set is gen-

erated using the lexical and phonetic distance features. Later, the candidates in the confusion set are ranked using a mixture of dictionary look up, word similarity based on lexical edit distance, phonemic edit distance, prefix sub-string, suffix sub-string and longest common subsequence (LCS), as well as context support metrics.

Gouws *et al.*, 2011 on the other hand, proposed an approach that depended highly on user-centric information such as the geological location of the users and the twitter client that the tweet is received from [3]. Using contextual metrics they modeled the transformation distributions.

Liu *et al.*, 2012 proposed a broad coverage normalization system, which integrates an extended noisy channel model, that is based on enhanced letter transformations, visual priming, string and phonetic similarity [16]. They try to improve the performance of the top n normalization candidates by integrating human perspective modeling. Yang and Eisenstein, 2013 introduced an unsupervised log linear model for text normalization [15]. Their joint statistical approach uses local context based on language modeling and surface similarity. Along with dictionary based models, Yang and Eisenstein's model have obtained a significant improvement on the performance of text normalization systems.

Hassan and Menezes, 2013 generated a normalization equivalence lexicon using Markov random walks on a contextual similarity lattice [9]. Our approach is different from theirs in several ways. First, our system makes use of the context of the OOV word in the input text, whereas their system is a dictionary-based method that always produces the same normalization to a given OOV word, regardless of its context in the input text. Besides the tokens themselves, we make use of the POS tags in creating the graph as well as the relative positions of the words in the social media text. Hassan and Menezes, 2013 create a bipartite graph, that is relatively more conservative in modeling the context of words. Context of a word is modeled as a window of words of size five. That is, two words to the right of a word and two words to the left of a word constitute the context of a word together. Even if one word is not the same, the context is considered to be different. On the other hand, in our graph, each neighboring token

contributes to the context information of a word, which leads to both a higher recall and a higher precision. One other difficulty in their approach that it requires a big clean corpus of English sentences, which is only avaliable commercially.

3. METHODOLOGY

In this thesis, we propose a graph based approach that models both contextual and lexical similarity features among an OOV word that requires normalization and candidate IV words. A high level overview of our system is shown in Figure 3.1. An input text is first preprocessed by tokenizing and Part-Of-Speech (POS) tagging. If the text contains an OOV word, the normalization candidates are chosen by making use of the contextual features which are extracted from a pre-generated directed word association graph, as well as lexical similarity features. Lexical similarity features are based on edit distance, longest common subsequence ratio, and double metaphone distance. In addition, a slang dictionary is used as an external resource to enrich the normalization candidate set. The details of the approach are explained in the following sub-sections.

3.1. Preprocessing

Tokenization is the first step in our system. It is the process of breaking the text into tokens, which are the smallest meaningful elements such as numbers, symbols, and emoticons. After tokenization, the next step in the pipeline is Part-of-Speech (POS) tagging each token using a POS tagger specifically designed for social media text. Unlike the regular POS taggers designed for well-written newswire-like text, social media POS taggers provide a broader set of tags specific to the peculiarities of social text [1, 2]. Using this extended set of tags we can identify tokens such as discourse markers (e.g. rt for retweets, cont. for a tweet whose content follows up in the coming tweet) or URLs. This enables us to better model the context of the words in social media text. A sample preprocessed sentence is shown in Table 3.1.

Table 3.1. Sample tokenized, POS tagged sentence (L: nominal+verbal, V: verb, D: determiner, N: noun, P: Preposition, A: adjective, C: punctuation).

Let's$_L$	start$_V$	this$_D$	morning$_N$	w$_P$	a$_D$	beatiful$_A$	smile$_N$	.$_C$

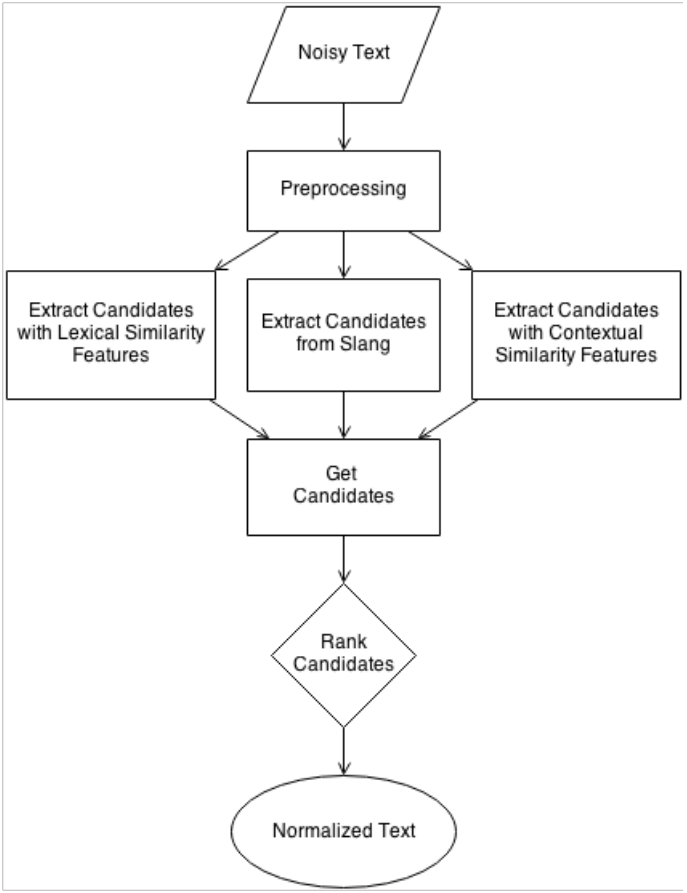

Figure 3.1. High level overview of our system.

As shown in Table 3.2, after preprocessing, each token is assigned a POS tag with a confidence score between 0 and 1. Later, we use these confidence scores in calculating the edge weights in our context graph. Note that even though the words *w* and *beatiful* are misspelled, they are tagged correctly by the tagger, with lower confidence scores though.

Table 3.2. Sample POS tagger output obtained by using CMU Ark Tagger [1, 2].

Token	*POS tag*	Tag confidence
with	*Preposition*	0.9963
a	*Determiner*	0.9980
beautiful	*Adjective*	0.9971
smile	*Noun*	0.9712

Token	*POS tag*	Tag confidence
w	*Preposition*	0.7486
a	*Determiner*	0.9920
beatiful	*Adjective*	0.9733
smile	*Noun*	0.9806

3.2. Graph construction

Contextual information of words is modeled through a word association graph created by using a large corpus of social media text. The graph encodes the relative positions of the POS tagged words in the text with respect to each other. After preprocessing, each text message in the corpus is traversed in order to extract the nodes and the edges of the graph. A node is defined with four properties: *id, oov, freq* and *tag*. The token itself is the *id* field. The *freq* property indicates the node's frequency count in the dataset. The *oov* field is set to True if the token is an OOV word. Following the prior work by Han and Baldwin, 2011 we used the GNU Aspell dictionary (v0.60.6) to determine whether a word is OOV or not [8]. We also edited the output of Aspell dictionary to accept letters other than "a" and "i" as OOV words. A portion of the graph that covers parts of the sample sentence in Table 3.1 is shown in Figure 3.2.

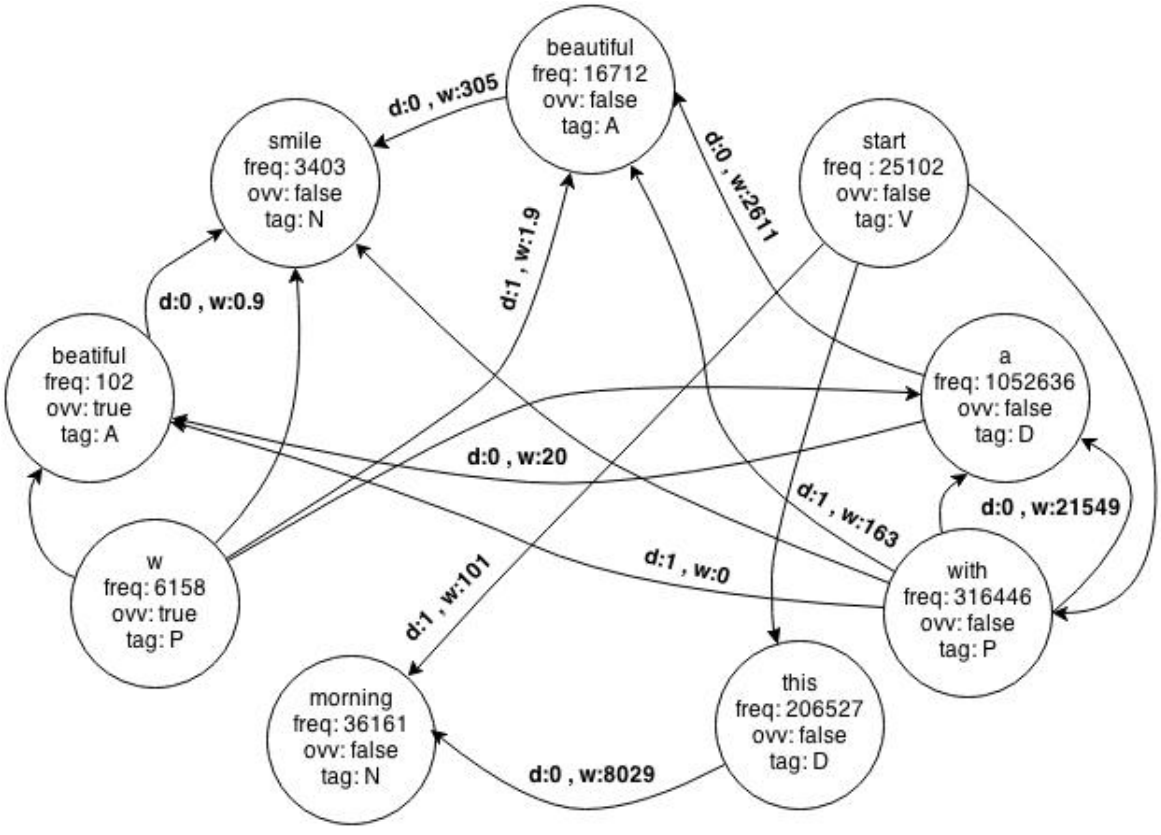

Figure 3.2. Portion of the word association graph for part of the sample sentence in Table 3.1. (d: distance, w: edge weight).

In the created word association graph, each node is a unique set of a token and its POS tag. This helps us to identify the candidate IV words for a given OOV word by considering not only lexical and contextual similarity, but also grammatical similarity in terms of POS tags. For example if the token *smile* has been frequently seen as a Noun or a Verb, and not in other forms in the dataset (e.g. Table 3.3), this provides evidence that it is not a good IV candidate as a normalization for an OOV token that has been tagged as a Pronoun. On the other hand, *smile* can be a good candidate for a Noun or a Verb OOV token, if it is lexically and contextually similar to it.

Table 3.3. The different nodes in the word association graph representing the token *smile* tagged with different POS tags.

node id	**freq**	**oov**	**tag**
smile	3	False	A
smile	3403	False	N
smile	2796	False	V

An edge is created between two nodes in the graph, if the corresponding word pair (i.e. token/POS pair) are contextually associated. Two words are considered as contextually associated if they satisfy the following criteria:

- The two words co-occur within a maximum word distance of $t_{distance}$ in a text message in the corpus.
- Each word has a minimum frequency of $t_{frequency}$ in the corpus.

Figure 3.3. The directionality of the edges is based on the sequence of words in the text messages in the corpus.

The directionality of the edges is based on the sequence of words in the text messages in the corpus (see Figure 3.3). In other words, an edge between two nodes is directed from the earlier seen token towards the later seen token. For example, Table 3.4 and Figure 3.4 show the edges that would be derived from a text including the phrase "with a beautiful smile". The *from* property indicates the first word and *to* is the latter in the phrase. The direction and the distance together represent a unique triplet. For each pair of nodes with a specific distance there is an edge with a positive weight, if the two nodes are related. Each co-occurrence of two related nodes increases the weight of the edge between them with an average of the nodes' POS tag confidence scores in the text message considered. If we are to expand the graph with the example phrase shown in Table 3.4, the weight of the edge with distance 2 from the node *with*$|P$ to the node *smile*$|N$ would increase by $(0.9963 + 0.9712)/2$, since the confidence score of the POS tag for the token *with* is 0.9963 and the confidence score of the POS tag of the token *smile* is 0.9712 as shown in Table 3.2.

Table 3.4. Example edges extracted from the sample phrase "with a beautiful smile".

from	**to**	**distance**	**weight**
with|P	smile|N	2	89
a|D	smile|N	1	274
beautiful|A	smile|N	0	305

3.3. Graph Based Contextual Similarity

Our graph based contextual similarity method is based on the assumption that an IV word that is the canonical form of an OOV word appears in the same context with the corresponding OOV word. In other words, the two nodes in the graph share several neighbors that co-occur within the same distances to the corresponding two words in social media text. We also assume that an OOV word and its canonical form should have the same POS tag.

Given an input text for normalization, the next step after preprocessing is finding the normalization candidates for each OOV token in the input text. For each ill-formed

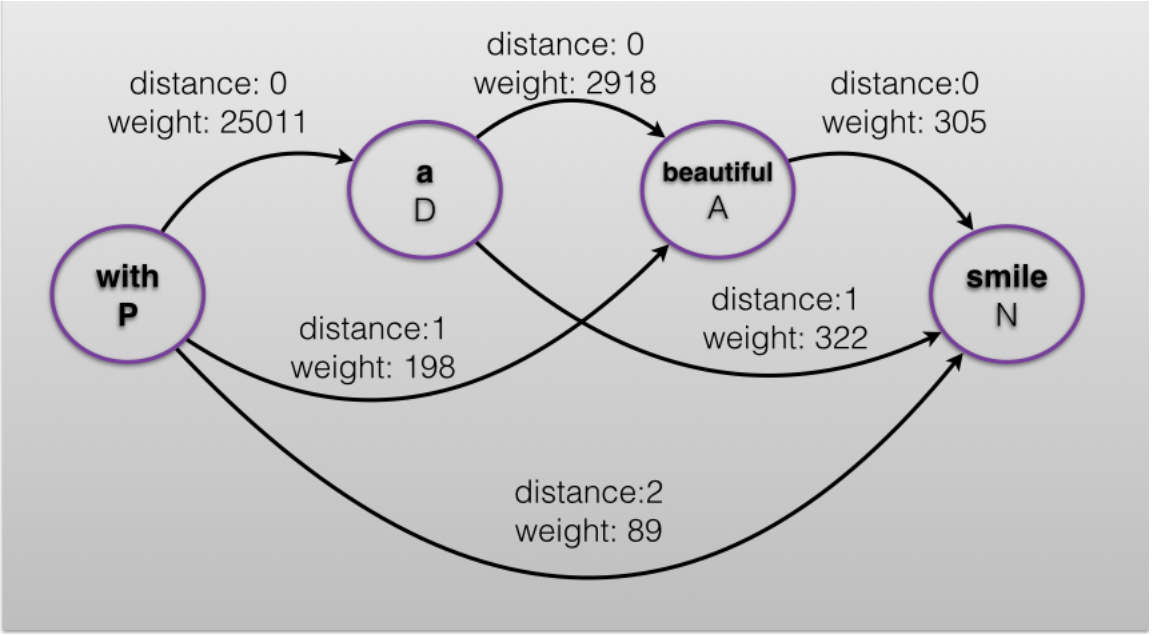

Figure 3.4. Sample nodes and edges from the word association graph.

OOV token o_i in the input text, first the list of tokens that co-occur with o_i in the input text and their positional distances to o_i are extracted. This list is called the neighbor list of token o_i, i.e., NL(o_i). Table 3.5 shows a sample neighbor list for the OOV token beatiful|A from the sample sentence in Table 3.1.

Table 3.5. Example neighbor list for the OOV node beatiful|A.

id	**tag**	**position**
w	P	-2
a	D	-1
smile	V	1

For each neighbor node n_j in NL, the word association graph is traversed, and the edges from or to the node n_j are extracted. The resulting edge list EL(o_i) has edges in the form of (n_j, c_k) or (c_k, n_j), where c_k is a candidate canonical form of the OOV word o_i. Here the neighbor node n_j can be an OOV node, but the candidate node c_k is chosen among the IV nodes. The edges in EL(o_i) are filtered by the relative distance of n_j to o_i as given in the NL(o_i). Any edge between n_j and c_k, whose distance is not the same as the distance between n_j and o_i is removed.

In addition to distance based filtering, POS tag based filtering is also performed on the edges in EL(o_i). Each candidate node should have the same POS tag with the corresponding OOV token. For the OOV token o_i that has the POS tag T_i, all the edges that include candidates with a tag other than T_i are removed from the edge list EL(o_i). Thus, EL(o_i) only contains edges where candidate nodes are tagged as T_i.

Figure 3.5 represents a portion from the graph where you can see the neighbours and candidates of the OOV node "beatiful". In the sample sentence in Table 3.1 there is two OOV token to be normalized, $o_1 = w$ and $o_2 = beatiful$. The neighbour list of o_2, NL(o_2) includes $n_1 = with$, $n_2 = a$ and $n_3 = smile$. For each neighbor in the NL(o_2), the candidate nodes ($c_1 = broken$, $c_2 = nice$, $c_3 = new$, $c_4 = beautiful$, $c_5 = big$, $c_6 = nice$, $c_7 = great$) are extracted. As shown in Figure 3.5, there are 11 lines representing the edges between the neighbors of the OOV token and the candidate nodes. These are representative edges in the EL(o_2). Each member of the edge list has the same tag (A for Adjective) as the OOV node "beatiful" and each has the same distance to the neighbor node they are connected as the OOV node. We are simply looking for the best replacements using the distance and POS tag properties.

Each edge in EL(o_i) consists of a neighbor node n_j, a candidate node c_k and an edge weight edgeWeight(n_j, c_k). The edge weight represents the likelihood or the strength of association between the neighbor node n_j and the candidate node c_k. As described in the previous section the edge weights are computed based on the frequency of co-occurrence of two tokens, as well as the confidence scores of their POS tags.

The edge weights of the edges in EL(o_2) are shown in Figure 3.5. The edges that are connected to the OOV neighbor "w" have smaller edge weights such as 3,5,26. On the other hand, the edges that are connected to common words have higher edge weight (e.g. the edge weight of the edge between nodes "a" and "new" is 24388). This indicates that those words are non OOV and more common words, and they co-occur very often in the same form ("a new").

Although this edge weight metric is reasonable for identifying the most likely

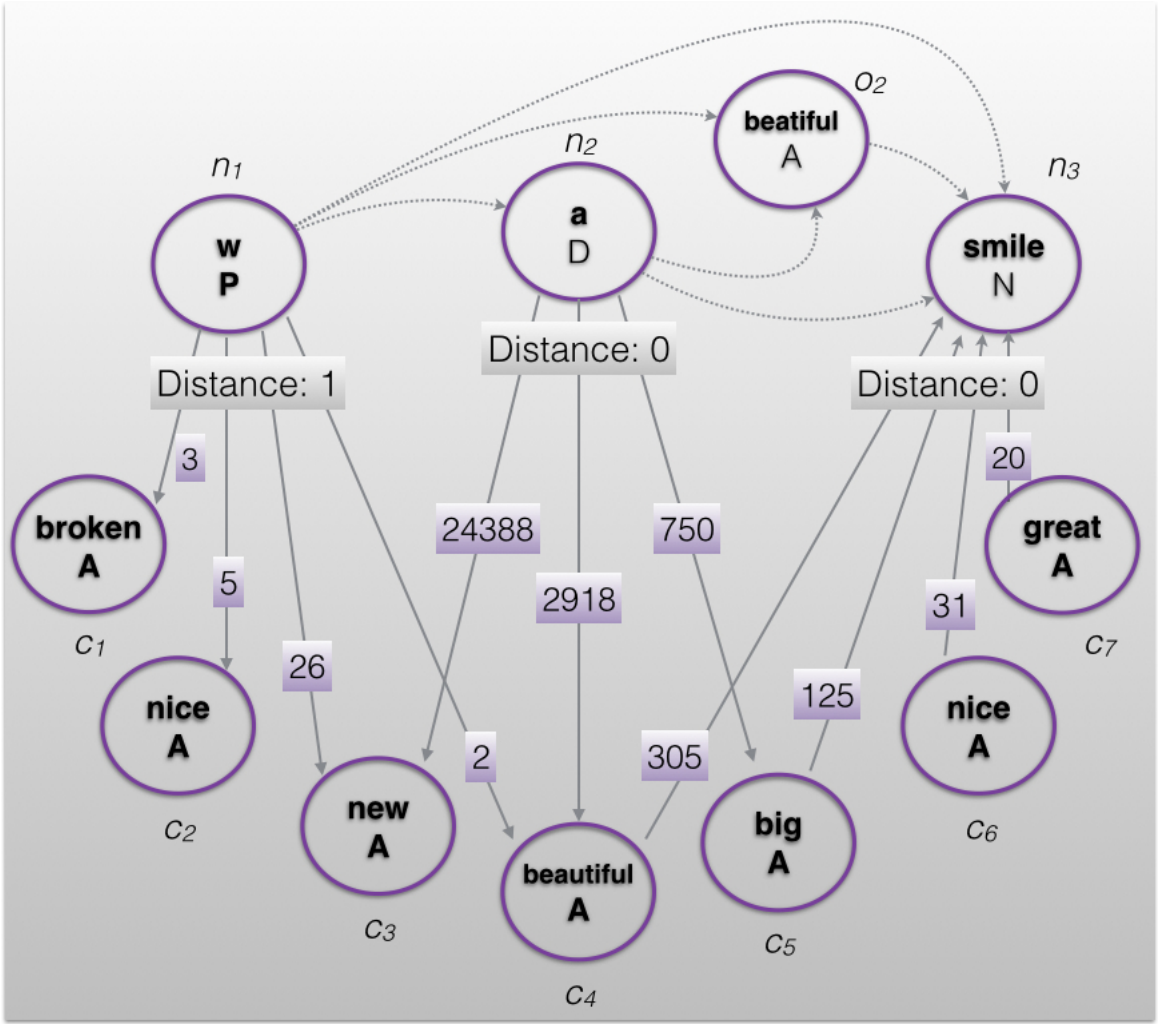

Figure 3.5. A portion of the graph that includes the OOV token "beatiful", its neighbors and the candidate nodes that each neighbor is connected to. Thick lines shows the edge list with relative weights.

canonical form for the OOV word o_i, it has the drawback of favoring words with high frequencies like these common words or stop words. Therefore, to avoid overrated words and get contextually relative candidates, we normalize the edge weight edgeWeight(n_j, c_k) with the frequency of the candidate node c_k as shown in Equation 3.1.

$$edgeWeightNormalized(n_j, c_k) = edgeWeight(n_j, c_k)/frequency(c_k) \tag{3.1}$$

Equation 3.1 provides a metric that captures contextual similarity based on binary associations. In order to achieve a more comprehensive contextual coverage, a contextual similarity feature is built based on the sum of the binary association scores of several neighbors. As shown in Equation 3.2, for a candidate node c_k the total edge weight score is the sum of the normalized edge weight scores edgeWeightNormalized(n_j, c_k), which are the edge weights coming from the different neighbors of the OOV token o_i. We expect this contextual similarity feature to favor and identify the candidates which are (i) related to many neighbors, and (ii) have a high association score with each neighbor.

$$\text{edgeWeightScore}(o_i, c_k) = \sum_{(n_j,c_k)\text{or}(c_k,n_j)\in \text{EL}(o_i)} \text{edgeWeightNormalized}(n_j, c_k) \qquad (3.2)$$

Figure 3.6 includes the top three candidates sorted by their scores. The candidate node "beautiful", c_4, has a frequency of 17900. Both three neighbors were paired in the edge list with it with the weights 2, 2918 and 305 respectively (Figure 3.5). After normalizing the edge weight by the frequency of the candidate node c_4, sum of those normalized weights gives us the edge weight score of the candidate node: edgeWeightScore$(o_2, c_4) = 0.18$.

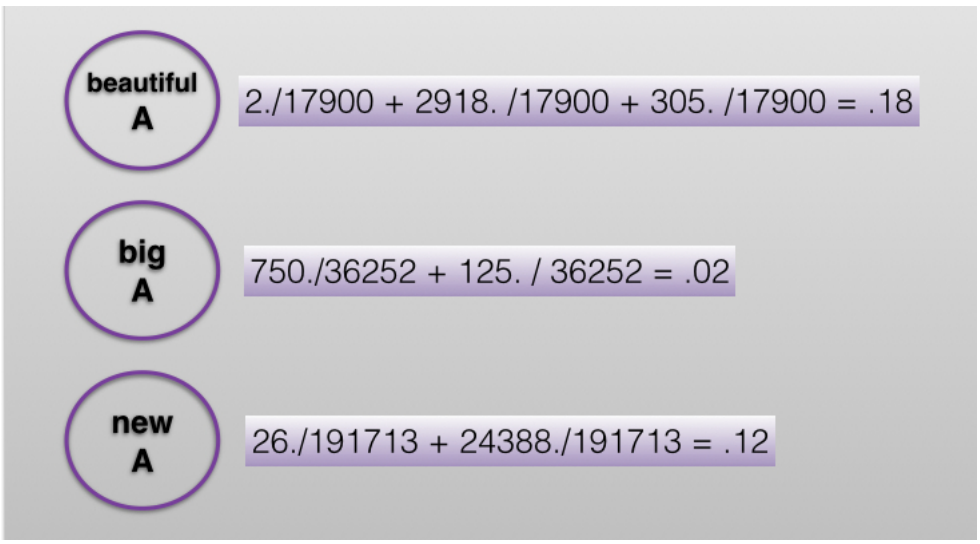

Figure 3.6. Candidates for "beatiful" sorted by their edge weight scores.

Our word association graph includes both OOV and IV tokens, and our OOV detection depends on the spellchecker which fails to identify some OOV tokens that have the same spelling with an IV word. In order to propose better canonical forms, the frequencies of the normalization candidates in the social media corpus have also been incorporated to the contextual similarity feature. Nodes with higher frequencies lead to tokens that are in their most likely grammatical forms.

The final contextual similarity of the token o_i and the candidate c_k is the weighted sum of the total edge weight score and the frequency score of the candidate (See Equation 3.3). The frequency score of the candidate is a real number between 0 and 1. It is proportional to the frequency of the candidate with respect to the frequencies of the other candidates in the corpus. Since the total edge weight score is our primary contextual resource, we may want to favor edge weight score. We give the frequency score a weight $0 \leq \beta \leq 1$ to be able to limit its effect on the total contextual similarity score.

$$\text{contextSimScore}(o_i, c_k) = \text{edgeWeightScore}(o_i, c_k) + \beta * \text{freqScore}(c_k) \tag{3.3}$$

Hereby, we have the candidate list $\text{CL}(o_i)$ for the OOV token o_i that includes all the unique candidates in $\text{EL}(o_i)$ and their contextual similarity scores calculated.

3.4. Lexical Similarity

Following the prior work in [8, 9], our lexical similarity features are based on edit distance [17], double metaphone (phonetic edit distance) [18], and a similarity function [19] which is based on Longest Common Subsequence Ratio (LCSR) [20].

Edit distance or in other words Levenshtein distance between two words is defined as the minimum number of single character changes such as insertion, deletion and substitution to convert one word into another. Edit distance has major application in

NLP especially in the spell checking.

Double metaphone is an extended word edit distance measure, that not only considers the characters but also the English pronunciations of the words. Like the Soundex algorithm, metaphone algorithm [21] encodes words. Similar sounding words, in other words phoneticly similar words shares the same keys in these encodings. Metaphone is an extended and more accurate version of Soundex. Double metaphone came out 10 years later than the metaphone and is called the second generation of the metaphone algorithm. Double metaphone produces two keys instead of one, presenting a more valid coverage for some irregularities.

The similarity cost function we are using is defined by Contractor *et al.*, 2010 [19], as the ratio between the LCSR of two words and the Edit Distance (ED) between their skeletons (the skeleton of a word is obtained by removing its vowels). The LCSR and the cost function are shown in Equation 3.4 and Equation 3.5. From now on we will refer to the simCost in Equation 3.5 as LCSR_ED.

$$\mathrm{LCSR}(o_j, c_k) = \frac{length(\mathrm{LCS}(o_j, c_k))}{\max(\mathrm{length}(o_j), \mathrm{length}(c_k))} \tag{3.4}$$

$$\mathrm{simCost}(o_j, c_k) = \frac{\mathrm{LCSR}(o_j, c_k)}{\mathrm{ED}(o_j, c_k)} \tag{3.5}$$

Following the tradition that is inspired from [22] before lexical similarity calculations, any repetitions of characters three or more times in OOV tokens are reduced to two (e.g. *goooood* is reduced to *good*). Then, the edit distance, phonetic edit distance, and LCSR_ED between each candidate in $\mathrm{CL}(o_i)$ and the OOV token o_i are calculated. Edit distance and phonetic edit distance are used to filter the candidates. Any candidate in $\mathrm{CL}(o_i)$ with an edit distance greater than t_{edit} and phonetic edit distance greater than $t_{phonetic}$ to o_i has been removed from the candidate list $\mathrm{CL}(o_i)$.

For the remaining candidates, the total lexical similarity score (Equation 3.6) is calculated using LCSR_ED and edit distance score[1] . Similar to contextual similarity

[1]an approximate string comparison measure (between 0.0 and 1.0) using the edit distance

score, here we have one main lexical similarity feature and one minor lexical similarity feature. The major lexical similarity feature is LCSR_ED and edit distance score is the minor. We assigned a weight $0 \leq \lambda \leq 1$ to the edit distance score to be able to lower its contribuiton while calculating the total lexical similarity score.

$$\text{lexSimScore}(o_i, c_k) = \text{LCSR\_ED}(o_i, c_k) + \lambda * \text{editDistScore}(o_i, c_k) \tag{3.6}$$

3.5. External Score

Since some social media text messages are extremely short and contain several OOV words, they do not provide sufficient context, i.e., IV neighbors, to enable the extraction of good candidates from the word association graph. Therefore, we extended the candidate list obtained through contextual similarity as described in the previous section, by including all the tokens in the word association graph that satisfy the edit distance and phonetic edit distance criteria. We also incorporated candidates from external resources, in other words from a slang dictionary and a transliteration table of numbers and pronouns (Table 3.6). If a candidate occurs in the slang dictionary or in the transliteration table as a correspondence to its OOV word, it is assigned an external score of 1, otherwise it is assigned an external score of 0.

The transliterations are first used in Gouws *et al.*'s work [3]. The transliteration table they have built has only numbers in common with ours. Besides the token and its transliteration we also use its POS tag information, which was not available in their system (Table 3.6).

The external score favors the well known interpretations of common OOV words. However unlike the dictionary based methodologies, our system does not return the corresponding unabbreviated word in the slang dictionary or in the transliteration table directly. Only an external score gets assigned and the candidate still needs to compete

https://sourceforge.net/projects/febrl/

Table 3.6. Transliteration Candidates extended from [3].

token	**tag**	**Transliteration**
1	“$”	“one”
2	“$”	“two”
3	“$”	“three”
4	“$”	“for”
5	“$”	“five”
6	“$”	“six”
7	“$”	“seven”
8	“$”	“eight”
9	“$”	“nine”
0	“$”	“zero”
2	“P”	“to”
“w”	“P”	“with”
“im”	“L”	“I’m”
“cont”	“~”	“continued”

with other candidates which may have higher contextual similarities and one of those contextually more similar candidates may be returned as the correct normalization instead of the candidate found equivalent to the OOV word in the slang dictionary (or in the transliteration table).

Some example entries from slang dictionary are shown in Table 3.7. The first and third examples in the table present why choosing directly the equivalent word in the dictionary as the correct normalization is a bad idea. The first word in the table is “2”. It is widely used as an abbreviation for the word “too”, however it is also used in place of the word “to”. Similarly “bc” can be normalized as “back” or “because” depending on the context of the sentence. That is why we do not return the corresponding candidate words directly but only increase their score using the external score metric. The other examples in Table 3.7 may give some idea about the overall content of the slang dictionary.

Table 3.7. Some entries from the slang dictionary at http://www.noslang.com/.

slang	**correspondence**
“2”	“too”
“acc”	“account”
“bc”	“ because”
“cr”	“can’t remember”
“dmi”	“don’t mention it”
“dupe”	“duplicate”
“gfx”	“graphics”
“h/e”	“however”
“h4kz0r5”	“hackers”
“iag”	“it’s all good”
“indie”	“independent”
“j00”	“you”
“nemore”	“anymore”
“y”	“why”

3.6. Overall Scoring

As shown in Equation 3.7, the final score of a candidate IV token c_k for an OOV token o_i is the sum of its lexical similarity score, contextual similarity score and external score with respect to o_i.

$$\begin{aligned} \text{candScore}(o_i, c_k) = \ & \text{lexSimScore}(o_i, c_k) + \text{contextSimScore}(o_i, c_k) \\ & + \text{externalScore}(o_i, c_k) \end{aligned} \tag{3.7}$$

4. EXPERIMENTS

4.1. Data sets

We used the LexNorm1.1 dataset [8] and Pennell *et al.*'s trigram dataset [14,23] to evaluate our proposed approach. LexNorm1.1 contains 549 tweets with 1184 manually annotated ill-formed OOV tokens. It has been used by recent text normalization studies for evaluation, which enables us to directly compare our performance results with results obtained by the recent previous work. The trigram dataset on the other hand is an SMS-like corpus collected from twitter status updates sent via SMS. The dataset does not include the complete tweet text but trigrams from tweets and one OOV word in each trigram is annotated. In total 4661 twitter status messages and 7769 tokens are annotated, previous work has used 80 % for training and the rest as test set, similarly we used the same 20 % of the dataset as test set to be able to report our results on the same basis as previous work.

4.2. Graph Generation

We used a large corpus of social media text to construct our word association graph. We extracted 1.5 GB of English tweets from Stanford's 476 million Twitter Dataset [24]. The language identification of tweets was performed by using the langid.py Python library [25, 26].

CMU Ark Tagger, which is a social media specific POS tagger achieving an accuracy of 95% over social media text [1, 2], is used for tokenizing and POS tagging the tweets. Besides the standard POS tags, the POS tagset of the Ark Tagger includes some extra POS tags specific to social media including URLs and emoticons; Twitter hashtags (#); and twitter at-mentions (@). One other tag that is special to social media is "~" which means the token is specific to a discourse function of twitter such as *rt, cont.*. Lastly G stands for miscellaneous words including multi word abbreviations like btw (by the way), nw (no way), and smh (somehow).

We made use of these social media specific tags to disambiguate some OOV tokens. For example if OOV token “cont” is tagged with the discourse function tag G, we added “continued” to the candidate list as an external node.

After tokenization, we removed the tokens that were POS tagged as mention (e.g. @brendon), discourse marker (e.g. RT), URL, email address, emoticon, numeral and punctuation. The remaining tokens are used to build the word association graph. After constructing the graph we only kept the nodes with a frequency greater than 8. For the performance related reasons, the relatedness thresholds $t_{distance}$ and $t_{frequency}$ were chosen as 3 and 8, respectively. The resulting graph contains 105428 nodes and 46609603 edges.

4.3. Candidate Set Generation

While extending the candidate set with lexical features we use $t_{edit} \leq 2 \vee t_{phonetic} \leq 1$ to keep up with the settings in Han *et al.* [8]. In other words, IV words that are within 2 character edit distance of a given OOV word or 1 character edit distance of a given OOV word under phonemic transcription were chosen as lexical similarity candidates.

4.4. Evaluation Metrics

The main evaluation metrics we have been used in this thesis are precision, recall and F-Measure. We run our text normalization method on our two test sets that are introduced in Section 4.1 and calculated each metric regarding to following definitions.

Precision (Equation 4.1) calculates the proportion of correctly normalized words among the OOV words that we could produce a normalization for. Recall (Equation 4.2) shows the amount of correct normalizations over the words that require normalization (ill-formaed OOV words). The main metric that we consider while evaluating the performance of our system is F-Measure (Equation 4.3) which is the harmonic mean of precision and recall values.

$$\text{Precision} = \frac{\text{\# of correctly normalized words}}{\text{\# of normalized words}} \tag{4.1}$$

$$\text{Recall} = \frac{\text{\# of correctly normalized words}}{\text{\# of words requiring normalization}} \tag{4.2}$$

$$\text{F-Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

5. RESULTS AND ANALYSIS

The results obtained by our proposed Contextual Word Association Graph (CWA-Graph) system on the LexNorm1.1 dataset, as well as the results of recent studies that used the same data set for evaluation are presented in Table 5.1.

Table 5.1. Results obtained on the LexNorm1.1 dataset.

Method	**Precision**	**Recall**	**F-measure**
Han and Baldwin, 2011	75.30	75.30	75.30
Liu *et al.*, 2011	84.13	78.38	81.15
Hassan and Menezes, 2013	85.37	56.40	69.93
Yang *et al.*, 2013	82.09	**82.09**	82.09
CWA-Graph	**85.50**	79.20	**82.20**

Our CWA-Graph approach achieves the best F-measure (82.20) and precision (85.50) among the recent previous studies. The high precision value is obtained without compromising much from recall (79.20). Our recall is the second best among others. The F-score (82.09) obtained by Yang *et al.*'s system is close to ours and the second best F-score, which on the other hand, has a lower precision than our approach [15].

Table 5.2 show the results of our system on the trigram SMS-like dataset. Without any modification to our system or to the parameters, we were able to improve results of Pennell *et al.* [14].

Table 5.2. Results obtained on the trigram SMS-like dataset.

Method	**Precision**	**Recall**	**F-measure**
Pennell *et al.*, 2011	69.70	**69.70**	69.70
CWA-Graph	**77.5**	67.7	**72.3**

The earlier work we compare our system with, assumes that the words to be normalized are given in advance. We also made the same assumption. However unlike

other systems [8, 15, 16], our system does not propose a normalization if there are no candidates that are lexically similar, grammatically correct and contextually close enough. For this reason, we managed to achieve a higher precision compared to the other systems. Besides, we made sure that the candidates have a minimum similarity either contextual, lexical, external or some degree of each feature. The results shown at Table 5.3 and Table 5.4 show that our approach can obtain even higher values of precision by tuning the system threshold (i.e. the minimum score in Equation 3.7 to return a token as a candidate canonical form of an OVV token).

Table 5.3. Comparison of results for different threshold values on LexNorm1.1, the setup we have used for our other experiments is shown in bold.

Threshold	**Precision**	**Recall**	**F-measure**
≤ 1	81.2	80.8	81.0
1.1	81.5	80.8	81.2
1.2	82.2	80.7	81.4
1.3	83.7	80.2	81.9
1.4	84.2	80.0	82.0
1.5	**85.5**	**79.2**	**82.2**
1.6	88.8	75.1	81.4
1.7	91.1	72.8	80.9
1.8	92.3	67.6	78.0
2	94.1	56.4	70.5

We also test our system on different window sizes. The window size is defined by the number of total neighbours of an OOV word in the given text. When we run our system with a contextual association threshold $t_{distance} = 3$, which means two words are considered as contextually associated, if they are within a maximum word distance of 3 in the text, 3 words to the left and 3 words to the right are taken into account for finding contextually similar candidates. For example when we look at the first example in Table 5.5, the $t_{distance}$ is set to 3 and for the OOV word "w", the window size is 7.

Table 5.4. Comparison of results for different threshold values on trigram dataset, the setup we have used for our other experiments is shown in bold.

Threshold	**Precision**	**Recall**	**F-measure**
≤ 0.8	72.2	68.9	70.5
1	72.1	68.8	70.4
1.1	72.6	68.8	70.7
1.2	73.1	68.8	70.9
1.3	74.3	68.2	71.1
1.4	75.5	67.8	71.4
1.5	**77.5**	**67.7**	**72.3**
1.6	80.8	64.8	71.9
1.7	83.3	58.4	68.7
1.8	86.1	52.1	64.9
1.9	87.6	45.4	59.8
2.1	91.2	33.8	49.3

On the other hand, in the second example, OOV word "beatiful" has 3 neighbors on the left but only one neighbour on the right, ending up with a window size 5 when the maximum window size set to 7. Thus the window size is defined by our threshold is only a maximum value, since the OOV word may or may not have enough neighbors to fit the maximum window size. From now on we will refer to maximum window size as just window size.

Considering twitter's limit on message lenght and users' tendency of using url and long hastags but short texts, the optimal window size is as expected relatively small. As shown in Table 5.6, the system achives best results with a window size of 7. This is also the setup we have used for our experiments. With higher window sizes OOV tokens get context information from outer word phrases which include contextually less relevant words. Choosing a smaller window size shortens the execution time, but on the other hand this results in missing some important contextual information.

Table 5.5. Window size examples from our sample sentence from Table 3.1 for OOV words "w" and "beatiful" with $t_{distance} = 3$ and $t_{distance} = 2$.

Max window size	**Sentence**
7	Let's [start this morning **w** a beatiful smile] .
7	Let's start this [morning w a **beatiful** smile] .
5	Let's start [this morning **w** a beatiful] smile .
5	Let's start this morning [w a **beatiful** smile] .

As described in Section 3.3 and 3.4 both contextual and lexical similarity is calculated using two metrics, one major and one minor. The major contextual similarity metric is the edge weight score of the candidates and the major lexical similarity metric is the LCSR which is shown to perform good on finding lexical similarity [8, 9]. For the minor metrics frequency score and edit distance score we introduced β and λ parameters respectively to be able to tune their contribution to the overall ranking.

We made experiments with different λ and β values. The performance of the

Table 5.6. Comparison of results for different window sizes.

Window size	**Precision**	**Recall**	**F-measure**
3	85.3	79.0	82.0
5	85.6	79.1	82.2
7	**85.5**	**79.2**	**82.2**
9	85.2	79.0	82.0

system for different values of λ and β tested on Lexnorm1.1 is shown in Figure 5.1. The best results are obtained when $\beta = 0.5$ in the range of $0.3 \leq \lambda \leq 1$ for Lexnorm1.1 (Figure 5.1) and a narrower range of $0.8 \leq \lambda \leq 1$ for the trigram dataset.

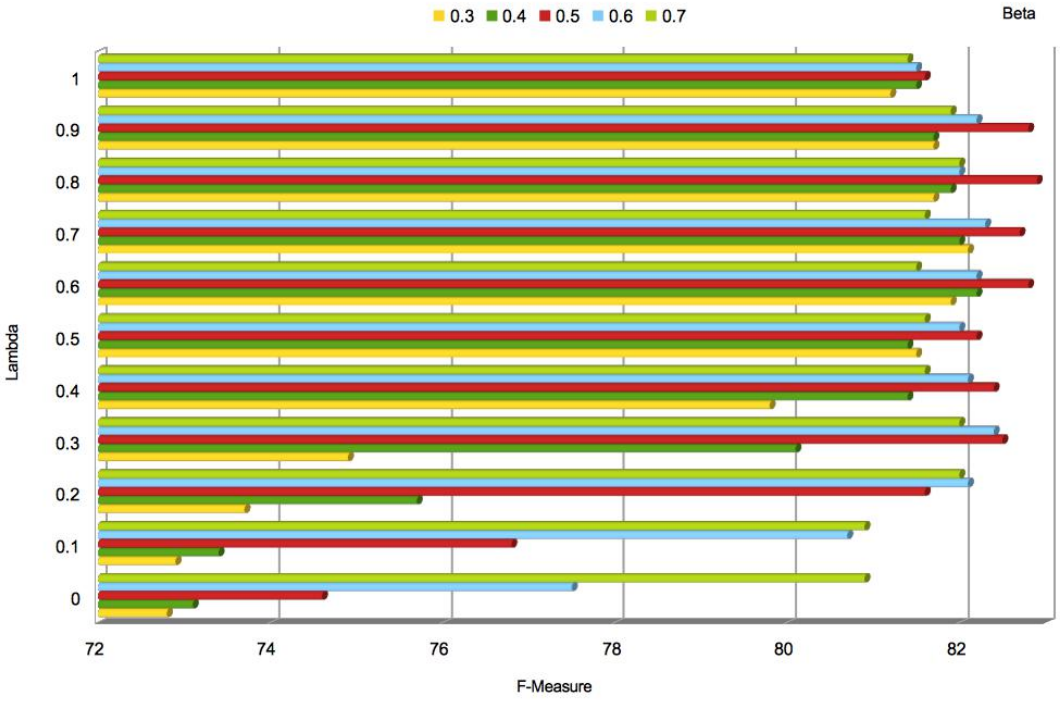

Figure 5.1. Results on LexNorm1.1 for different λ and β values.

We choose $\beta = 0.5$ and $\lambda = 0.5$ values in our main system. However it is possible to increase these results using higher λ values (Figure 5.2).

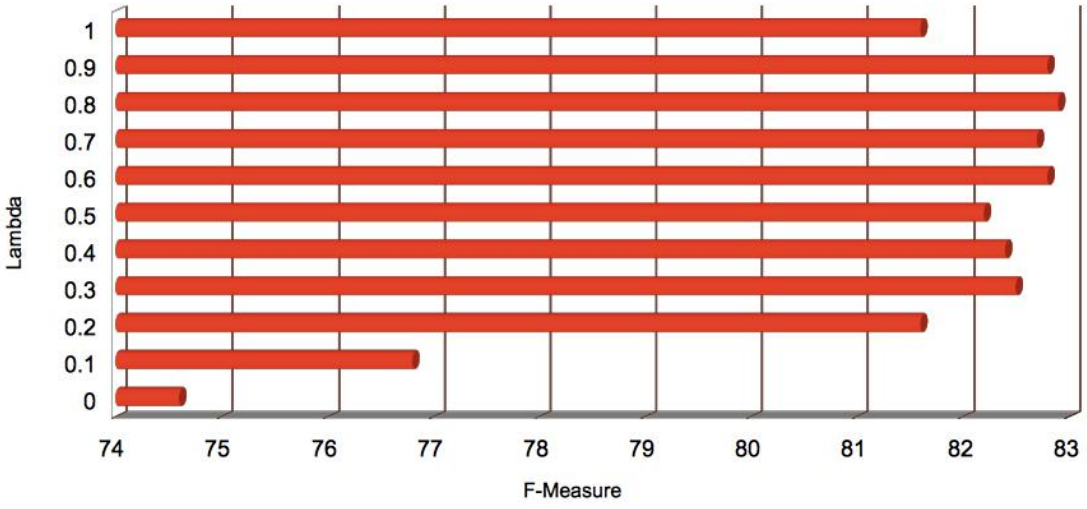

Figure 5.2. Results on LexNorm1.1 for $\beta = 0.5$ and $0 \leq \lambda \leq 1.0$.

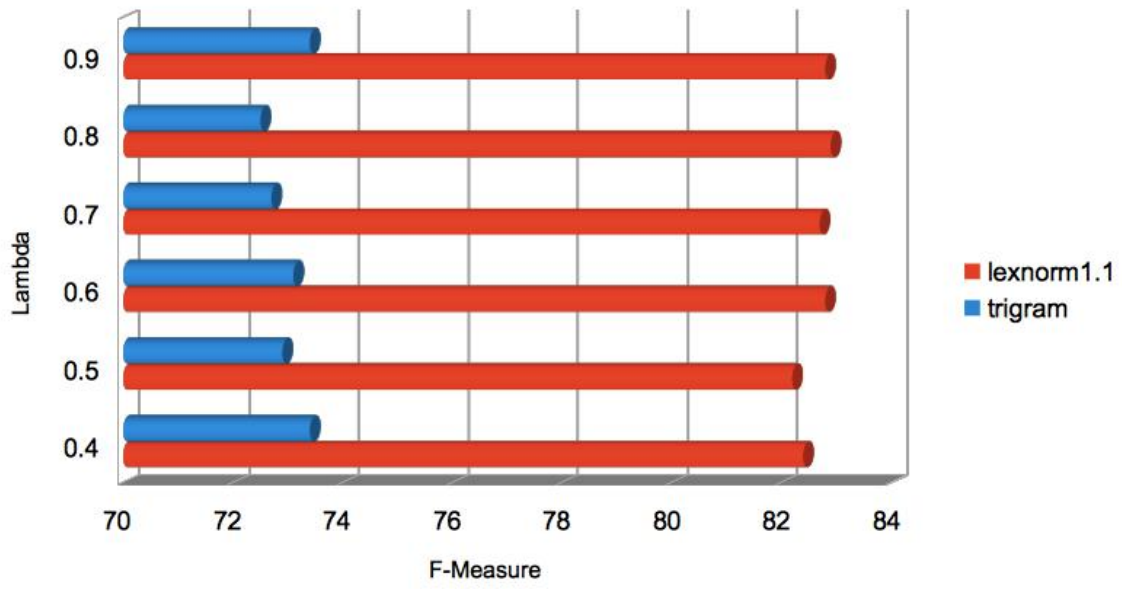

Figure 5.3. Results on LexNorm1.1 and trigram dataset for $\beta = 0.5$ and $0.4 \leq \lambda \leq 0.9$.

6. CONCLUSION AND FUTURE WORK

In this thesis, we present an unsupervised graph based approach for contextual text normalization. The proposed approach can analyze grammatical and contextual information from the noisy input text. The task of normalization is highly dependent on understanding and capturing the dynamics of the informal nature of noisy text. Our word association graph is built using a large unlabeled social media corpus. It helps to derive contextual and grammatical analysis on both clean and noisy data.

It is important to emphesize the difference between using corpus based contextual information and using contextual information of the input text (input context). We use corpus based contextual information for building our CWA-Graph. The graph encodes the context information of words with regard to other words they are contextually associated with. Given an input text that includes an OOV word to normalize, each neighbouring word in the input text gives us context information that we can associate the OOV word with candidates. We use this input based context information to find the correct normalization of the OOV word using contextual associations.

Using input context to find the correct normalization of OOV words is the major advantage of our system. Many other systems use the corpus based contextual information to find the normalizations, however this approach is led by statistical information, in other words it finds which IV word the OOV word is commonly normalized to. However, using input context to find normalizations helps us find the correct normalization, even if it is not the statistically dominant one. That way we can use statictical information to connect/associate the words and use input context to associate the correct normalization with the OOV word.

We compared our approach with the recent social media text normalization systems and achieved state-of-the-art precision and F-measure scores. We reported our results on two datasets. The first one was the standard text normalization dataset derived from Twitter. Our results on this dataset showed that our system can serve

as a high precision text normalization system which is highly preferable as an NLP preprocessing step [9]. Our system achieved over 94 % precision where the highest precision in the literature at the same recall level is 85.37 % [9].

The second dataset we tested our approach is a SMS-like trigram dataset. This tests showed that the CWA-Graph can perform good on SMS data as well. However the trigram nature of the dataset resulted in input texts which are very limited with regard to contextual information. Nevertheless, our system achieved over 72 % F-Measure using this contextual information even though it is limited because the SMS-like nature of the dataset makes it rich on abbreviations. Abbreaviations are difficult to normalize using only lexical features due to the higher edit distance values. Not filtering candidates with higher degree of lexical distances results in huge lists of candidates [8], which makes it harder to choose the right candidates. Overall, although being around 8 % lower than standard data set, the performance of our system on SMS-like data is reasonably high.

The two lexical metrics that we used (LCSR and edit distance) have already been shown to perform good on text normalization [8, 9]. However LCSR was a better approach than the simple edit distance score. Depending on this assumption we choose one major metric for both lexical and contextual similarity calculations and lowered the weight of the second metric (minor metric). When we run our system with different values of the minor metric parameters, the following observations are made.

- It is possible to increase the performance of the system by tuning the minor metrics.
- The system performs best for the values of weights that are closer to 0.5.
- The system performs worse when the weights are set to 0 (not using the minor features) and set to 1 (giving equal weight to the minor feature and major feature).

That showed that lowering the contribution of these minor metrics increases the overall performance of the normalization task both contextually and lexically, whereas, the absence of these minor metrics or making them also the major metrics by assigning

the weights as 1 leads to loss in performance.

Except for the double metaphone algorithm that encodes the phonetic similarities among words in English, the proposed approach is highly language independent. The system does not require a clean corpus or an annotated corpus, the CWA-Graph can be built by using the publicly avaliable social media text.

As future work, OOV detection can be added to the system. Another task to do next could be integrating the normalization system into an application. This way we can measure how well the normalization system performs as a preprocessing step in NLP applications. The last item in our future work list is adopting the system to be able to work on languages other than English.

REFERENCES

1. Owoputi, O., B. O'Connor, C. Dyer, K. Gimpel, N. Schneider and N. A. Smith, "Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters", *Proceedings of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pp. 380–390, 2013.

2. Gimpel, K., N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan and N. A. Smith, "Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments", *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, pp. 42–47, 2011.

3. Gouws, S., D. Metzler, C. Cai and E. Hovy, "Contextual Bearing on Linguistic Variation in Social Media", *Proceedings of the Workshop on Languages in Social Media*, pp. 20–29, 2011.

4. Choudhury, M., R. Saraf, V. Jain, A. Mukherjee, S. Sarkar and A. Basu, "Investigation and Modeling of the Structure of Texting Language", *International Journal on Document Analysis and Recognition*, Vol. 10, No. 3, pp. 157–174, 2007.

5. Eisenstein, J., "What to Do About Bad Language on the Internet", *Proceedings of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pp. 359–369, 2013.

6. Sproat, R., A. W. Black, S. Chen, S. Kumar, M. Ostendorf and C. Richards, "Normalization of Non-Standard Words", *Computer Speech & Language*, Vol. 15, No. 3, pp. 287–333, 2001.

7. Ritter, A., C. Cherry and B. Dolan, "Unsupervised modeling of twitter conversations", *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 172–180,

2010.

8. Han, B. and T. Baldwin, "Lexical Normalisation of Short Text Messages: Makn Sens a #Twitter", *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pp. 368–378, 2011.

9. Hassan, H. and A. Menezes, "Social Text Normalization Using Contextual Graph Random Walks", *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 1577–1586, 2013.

10. Brill, E. and R. C. Moore, "An Improved Error Model for Noisy Channel Spelling Correction", *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pp. 286–293, 2000.

11. Toutanova, K. and R. C. Moore, "Pronunciation Modeling for Improved Spelling Correction", *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 144–151, 2002.

12. Cook, P. and S. Stevenson, "An Unsupervised Model for Text Message Normalization", *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pp. 71–78, 2009.

13. Aw, A., M. Zhang, J. Xiao and J. Su, "A Phrase-based Statistical Model for SMS Text Normalization", *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 33–40, 2006.

14. Pennell, D. and Y. Liu, "A Character-Level Machine Translation Approach for Normalization of SMS Abbreviations.", *Fifth International Joint Conference on Natural Language Processing*, pp. 974–982, 2011.

15. Yang, Y. and J. Eisenstein, "A Log-Linear Model for Unsupervised Text Normal-

ization", *Proceedings of the Empirical Methods on Natural Language Processing*, pp. 61–72, 2013.

16. Liu, F., F. Weng and X. Jiang, "A Broad-Coverage Normalization System for Social Media Language", *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pp. 1035–1044, 2012.

17. Levenshtein, V. I., "Binary Codes Capable of Correcting Deletions, Insertions and Reversals", *Soviet Physics Doklady*, Vol. 10, p. 707, 1966.

18. Philips, L., "The Double Metaphone Search Algorithm", *C/C++ Users Journal*, Vol. 18, No. 6, pp. 38–43, Jun. 2000.

19. Contractor, D., T. A. Faruquie and L. V. Subramaniam, "Unsupervised Cleansing of Noisy Text", *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 189–196, 2010.

20. Melamed, I. D., "Bitext Maps and Alignment via Pattern Recognition", *Computational Linguistics*, Vol. 25, No. 1, pp. 107–130, 1999.

21. Philips, L., "Hanging on the Metaphone", *Computer Language*, Vol. 7, No. 12, 1990.

22. Kaufmann, M. and J. Kalita, "Syntactic Normalization of Twitter Messages", *Proceedings of the 8th International Conference on Natural Language Processing*, pp. 149–158, 2010.

23. Pennell, D. L. and Y. Liu, "Normalization of Informal Text", *Computer Speech & Language*, Vol. 28, No. 1, pp. 256 – 277, 2014.

24. Yang, J. and J. Leskovec, "Patterns of Temporal Variation in Online Media", *Proceedings of the Forth International Conference on Web Search and Web Data Mining*, pp. 177–186, 2011.

25. Lui, M. and T. Baldwin, "Langid.Py: An Off-the-shelf Language Identification Tool", *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 25–30, 2012.

26. Baldwin, T. and M. Lui, "Language Identification: The Long and the Short of the Matter", *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 229–237, 2010.