A Graph Based Approach for Contextual Text Normalization

Abstract

The informal nature of social media text render 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.

Keywords: Text Normalization, Twitter, micro-blogs, social media

1. Introduction

Social text has become an enormous part of our lives. We are moving towards to an era that we will be talking using machines more than we talk to each other. Social platforms make mass amounts of people communicate via typed or transcribed text. That is the era that the news are spreading digitally via social media other than edited newspaper articles.

With these in mind, it has been also starting of a new era for text analytics researches. The recent studies on social media such that Stock Prediction [1], politeness detection [2], disaster detection [3] tries to lighten up the road of this digitized future of ours.

Therefore analyzing social media text is a challenge for itself. Due to its noisy nature, many NLP tools are performing poorly on social media text [4]. The problems that noise in the social media text generates for NLP tools can be overcome by some preprocessing steps.

Unlike spoken and written language, digitized language has its own form and nature. Since the beginning of World Wide Web, internet has it's own slang. *lol* meaning *laughing out loudly*, *xoxo* meaning *kissing*, *4u* meaning *for you* are the oldest examples of this slang. Everyday new slangs as well as new words such as iTunes and new abbreviations are coming up. It is a huge, an evolving language that has long gone beyond the reach and control of spellcheckers and slang dictionaries.

Text normalization is a preprocessing step to restore noisy forms of text to its original(canonical) form [5] to make use in NLP applications or more broadly to understand the digitized text better. For example *talk 2 u later*

ppl	people		r	are
havent	haven't		mor	more
tmr	tomorrow		doin	doing
soooo	so		n	and
sooon	soon		friiied	fried
raight	right		finge	finger
raight	alright		kissin	kissing

Table 1: Example of noisy tokens and their normalized form

can be normalized as *talk to you later* or similarly *enormoooos, enrmss, enourmos* can be normalized as *enormous*. Those noisy tokens are referred as Out of Vocabulary(OOV) words. Normalization task restores OOV words to their In Vocabulary (IV) form.

However not every OOV word should be considered for normalization. The social text is continuously evolving with new words and named entities that are not in the vocabularies of the systems [6]. The OOV tokens that should be considered for normalization are referred to as ill-formed words. Oppositely an OOV word can sometimes lexically fit an IV word (Ex: *tanks* is both an IV word and OOV word with the canonical form *thanks*). The task of recognizing which tokens are OOV, and which of those are ill-formed are beyond the scope of this paper.

In [7] Choudhury et Al. proposes that the OOV words observed in noisy text can be classified into two groups, unintentional and intentional errors. The unintentional errors are caused by (1) pressing of the wrong key, (2) pressing of a key more than the desired number of times, (3) deletion of a character or (4) 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 paper we propose a new approach to text normalization. A graph based model, which benefits from both lexical, contextual and grammatical features of social text.

2. Related Work

Early work on text normalization mostly made use of noisy channel model. The first work that had a significant performance improvement over the previous research was Brill and Moore,2000 [8]. They proposed a novel noisy channel model for spell checking based on string to string edits. Their error model depended on probabilistic modelling of sub-string transformations. Toutanova et al.,2002 improved this approach by extending the error model with phonetic similarities over words [9].

Choudhury et al.,2007 developed a supervised Hidden Markov Model based approach for normalizing SMS Texts [7]. Cook and Stevenson,2009 has expanded this model by introducing an unsupervised noisy channel model [10]. 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 modelled explicitly. Aw et al.,2006 proposed a phrase-based statistical MT model for the task [11]. They defined the problem as translating the SMS language to English language.

What these methods are missing is that they do not consider contextual features and they observe each token that has an unique normalization. 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.

Also the supervised models require annotated data, which is not available for normalized text [19] or difficult to create.

More recent approaches handle the text normalization task by building normalization lexicons. Han et al.,2011 developed a two phased model where they only consider the OOV words for normalization [5]. First a confusion set is generated using the lexical and phonetic distance features. Latter the candidates in the confusion set is selected using a mixture of dictionary lookup, word similarity based on lexical edit distance, phonemic edit distance, prefix sub-string, suffix sub-string and longest common subsequence(LCS) and context support metrics.

Gouws et al.,2011 on the other hand, proposed an approach that depended highly on contextual information such as the geological location and depended on the users and twitter client that the tweet is received from [12]. Using contextual metrics they modelled the transformation distributions.

Liu et al.,2012 proposes broad coverage normalization system integrates an extended noisy channel model based on enhanced letter transformations, visual priming, string and phonetic similarity [22]. They try to improve the performance of top n normalization candidates by integrating human perspective modeling.

Hassan and Menezes,2013 generates a normalization equivalences lexicon using Markov random walks on a contextual similarity lattice [6]. One of the biggest difference between our systems are they make use a huge clean vocabulary.

Most recent work on text normalization is the unsupervised log linear model of Yang and Eisenstein,2013 [19]. Their joint statistical approach uses local context based on an LM 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. We believe several models such as the morphophonemic similarity, MT, Maximum Likelihood, etc. has their own limits. Higher performance text normalization systems should make use of contextual analysis.

3. Methodology

In this paper, 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 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.

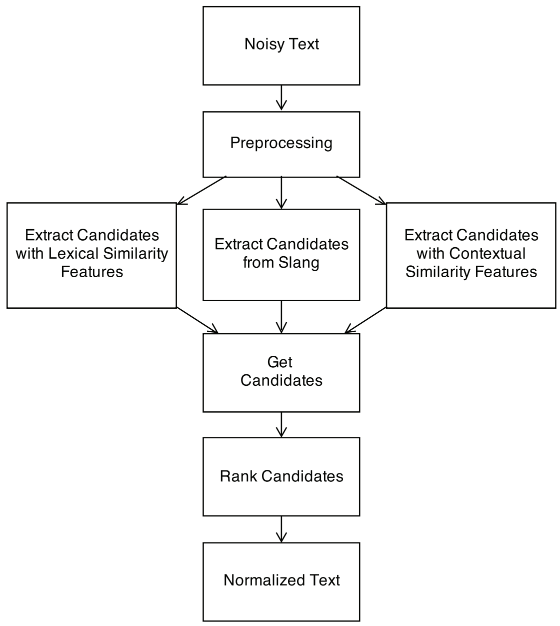

Figure 1: High level overview of our system

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 [13, 14].

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.

As shown in Table 2, after preprocessing, each token is assigned a POS tag with a confidence measure 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.

Token	POS tag	Confidence
with	P	0.9963
a	D	0.998
beautiful	A	0.9971
smile	N	0.9712

Token	POS tag	Confidence
w	P	0.7486
a	D	0.9920
beatiful	A	0.9733
smile	N	0.9806

Table 2: Sample POS tagger output obtained by using CMU Ark Tagger (P:Pronoun, D:Determiner, A:Adjective, N:Noun, G:Miscellaneous) [13, 14]

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, tag*. The token itself and it's POS tag form 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 [5]. Table 3 shows a sample tokenized and tagged sentence from the corpus, as well as the nodes and edges that are extracted from it to create the word association graph. A portion of the graph that covers this sample sentence is shown in Figure 2.

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

Tokens	Let's, start, this, morning, w, a, beatiful, smile, .
Nodes	Let's|L, start|V, this|D, morning|N, w|P, a|D, beatiful|A, smile|N, .|,
Edges	{Let's|L, start|V , distance:1},{Let's|L, this|D, distance:2}, ... {a|D, beatiful|A, distance:1}, {a|D, smile|N, distance:2}, {beatiful|A, smile|N, distance:1}

Table 3: Sample tokenized, POS tagged sentence and the corresponding nodes and edges in the word association graph.

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 4), this provides evidence that it is not a good IV

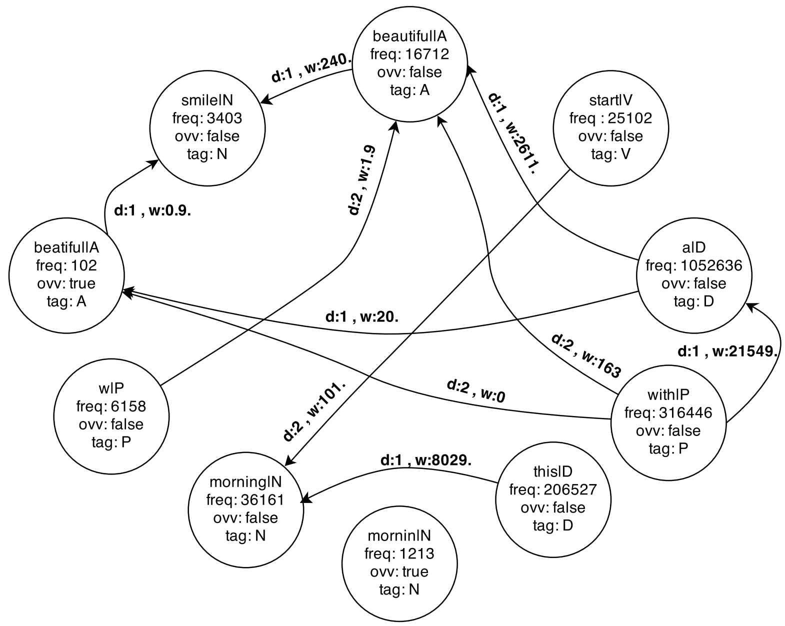

Figure 2: Sample nodes and edges from the word association graph.

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.

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 d_t in a text message in the corpus.
- Each word has a minimum frequency of f_t in the corpus.

The directionality of the edges is based on the sequence of words in the

node id : smile|A , freq : 3, oov : False, tag : A
node id : smile|N , freq : 3403, oov : False, tag : N
node id : smile|V , freq : 2796, oov : False, tag : V

Table 4: The nodes in the word association graph representing the token *smile* tagged with different POS tags.

text messages in the corpus. In other words, an edge between two nodes is directed from the earlier seen token towards the later seen token. For example, Table 5 shows 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 5, the weight of the edge with distance 3 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 2.

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

from : with|P, to : smile|N, dis : 3, weight : 72.24415
from : a|D, to : smile|N, dis : 2, weight : 274.37365
from : beautiful|A, to : smile|N, dis : 1, weight : 240.716

Table 5: Example edges extracted from the sample phrase “with a beautiful smile”

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 OOV token t_i in the input text, first the list of tokens that co-occur with t_i in the input text and their positional distances to t_i are extracted. This list is called the neighbor list of token t_i, i.e., NL_i. Table 6 shows a sample neighbor list for the OOV token beatiful|A from the sample sentence in Table 3.

w|P, position: -2
a|D, position: -1
smile|V, position: 1

Table 6: Example neighbor list for the OOV node beatiful|A

For each neighbor node n_{ij} in NL_i, the word association graph is traversed, and the edges from or to the node n_{ij} are extracted. The resulting edge list EL_{ij} has edges in the form of (n_{ij}, c_{ik}) or (c_{ik}, n_{ij}), where c_{ik} is a candidate canonical form of the OOV word t_i. Here the neighbor node n_{ij}

can be an OOV node, but the candidate node c_{ik} is chosen among the IV nodes. The edges in EL_{ij} are filtered by the relative distance of n_{ij} to t_i as given in the NL_i. Any edge between n_{ij} and c_{ik}, whose distance is not the same as the distance between n_{ij} and t_i is removed.

In addition to distance based filtering, POS tag based filtering is also performed on the edges in EL_{ij}. Each candidate node should have the same POS tag with the corresponding OOV token. For the OOV token t_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_{ij}. Thus, EL_{ij} only contains edges where the c_{ik} nodes are tagged as T_i.

Each edge in EL_{ij} consists of a neighbor node n_{ij}, a candidate node c_{ik} and an edge weight ew_{ijk}. The edge weight, represents the likelihood or the strength of association between the neighbor node n_{ij} and the candidate node c_{ik} (Eq 1). 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. Although this edge weight metric is reasonable for identifying the most likely canonical form for the OOV word t_i, it has the drawback of favoring words with high frequencies like the stop words. Therefore, we normalize the edge weight ew_{ijk} with the frequency of the candidate node c_{ik} as shown in Eq 2.

$$ew(n,p,c) = \begin{cases} w : (n, c, distance = |p|, weight = w), & \text{if } pos < 0 \\ w : (c, n, distance = |p|, weight = w), & \text{otherwise} \end{cases} \tag{1}$$

$$ewNormalized(n,p,c) = ew(n,p,c)/frequency(c) \tag{2}$$

Eq 2 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 Equations 3 and 4, for a candidate node c_{ik} the total edge weight score is the sum of the edge weight scores ew_{ijk}, which are the edge weights coming from the different neighbors of the OOV token t_i. We expect this contextual similarity feature to favor and identify the candidates which are (1) related to many neighbors, and (2) have a high association score with each neighbor.

$$contSimCostNeigh(t, n, c, p) = \sum_{n,c \in EL(t,n)} ewNormalized(n, p, c) \quad (3)$$

$$edgeWeightScoreNeigh(t, c) = \sum_{n,p \in NL(t)} contSimCostNeigh(t, n, c, p) \quad (4)$$

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 t and the candidate c is the weighted sum of the total edge weight score and the frequency score of the candidate (See Eq 5). 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, the weight of the frequency feature is set as half of the weight of the total edge weight score.

$$contScore(t,c) = \lambda_a edgeWeightScore(t,c) + \frac{\lambda_a}{2} freqScore(c) \qquad (5)$$

Hereby, we have the candidate list CL_i for the OOV token t_i that includes all the unique candidates in EL_i and their contextual similarity scores calculated.

3.4. Lexical Similarity

Following the prior work in [5, 6], our lexical similarity features are based on edit distance [15], double metaphone (phonetic edit distance) [16], and longest common subsequence ratio (LCSR) [17].

Following the tradition that is inspired from [18] 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 between each candidate in CL_{ij} and the OOV token t_i are calculated. Edit distance and phonetic edit distance are used to filter the candidates. Any candidate in CL_{ij} with an edit distance greater than ed_t and phonetic edit distance greater than ped_t to t_i has been removed from the candidate list CL_{ij}.

For the remaining candidates, the total lexical similarity score (Eq 6) is calculated using LCSR and edit distance score[1]. Since the main lexical

[1]an approximate string comparison measure (between 0.0 and 1.0) using the edit distance `https://sourceforge.net/projects/febrl/`

feature is LCSR, it is assigned twice the weight of the edit distance 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 7). If a token occurs in the slang dictionary or in the transliteration table it is assigned an external score of 1, otherwise it is assigned an external score of 0.

$$lexScore(t,c) = \lambda_a LCSR(t,c) + \frac{\lambda_a}{2} editDistScore(t,c) + \lambda_a externalScore(t,c) \tag{6}$$

As shown in Equation 7, the final score of a candidate IV token c for an OOV token t is the sum of its lexical similarity score and contextual similarity score with respect to t.

$$candScore(t,c) = lexScore(t,c) + contScore(t,c) \tag{7}$$

4. Experiments

4.1. Data set

We used LexNorm1.1 dataset [5] to evaluate our approach discussed in the methodology section. LexNorm1.1 contains 549 tweets with 1184 ill-formed

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

Table 7: Transliteration Candidates improved

OOV tokens.

4.2. Graph Generation

We used a large amount of social media text to construct our co-occurrence graph. We extracted 1.5GB tweets in English from Stanford’s 476 million Twitter Dataset [19]. The language identification of tweets was performed using the langid.py [20, 21].

After tokenization we removed tokens (that were) POS tagged as mention (@brendon), discourse marker (ex: RT), URL, email addresses, emoticons, numerals and punctuations. Remaining tokens are used to build the graph. ????

For tokenization and POS tagging the tweets, we used CMU Ark Tweet Tagger [13, 14]. Ark Tweet Tagger is a social media specific POS tagger and is reported to perform 95% accuracy over social media text.

The POS tagset of ark tagger includes some extra tags beside the standard part of speech tags that is specific to social media: URLs and emoticons; Twitter hashtags #; twitter at-mentions (@). One other tag that is special to social media is ˜ means the token specific to a discourse function of twitters. Lastly G stands for miscellaneous words including multi word abbreviations like btw (by the way), nw (no way), smh (somehow).

We made use of this 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 constructing the graph we only kept the nodes with a frequency greater than 8. For the performance related reasons, the relatedness thresholds d_t and f_t were chosen as 3 and 8 respectively. We had remaining 105428 nodes and 46609603 edges in the graph after the setup.

4.3. Candidate Set Generation

While extending the candidate set with lexical features we use $ed_t \leq 2 \vee ped_t \leq 1$ to keep up with the settings in Han et al. [5]. IV words that are within 2 character edit distance of given OOV word or 1 character edit distance of given OOV word under phonemic transcription were chosen as lexical similarity candidates.

4.4. Results and Analysis

The results are presented in Table 8. Our system, Contextual Word Association Graph (CWA-Graph), performed the highest precision among all other systems compared with 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 et al.,2013	85.37	56.40	69.93
Yang et al.,2013	82.09	82.09	82.09
CWA-Graph	86.20	78.0	82.0

Table 8: Empirical Results produced on LexNorm1.1 dataset

As well as precision, we managed to raise the F-Measure score without compromising much from the recall. Yet, we have the highest F-Measure score except the Yang et al.'s [19].

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 ([19, 22, 5]), our system may 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 recall compared to other systems. Besides, we made sure that the candidates have a minimum similarity either contextual, lexical, externally or some degree of each feature. In Table 9 you can see how we reached higher degrees of precision by fine tuning the system threshold.

5. Conclusion

We have presented an unsupervised graph based approach for contextual text normalization. We compared our work with the recent state of the art normalization systems and achived highest Precision and F-measure scores.

Threshold	Precision	Recall	F-measure
≤ 1	81.40	81.1	81.20
1.1	84.40	79	81.60
1.2	87.90	76.30	81.7
1.3	84.80	79.50	82.00
1.4	85.40	79.	81.90
1.45	86.20	78.1	82.
1.6	90.00	72.1	80.10
1.7	92.50	68.9	79.00
1.8	94.50	60.4	73.70

Table 9: Comparison of results for different threshold values, which set as at least λ_a

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 informal nature of noisy text. Our word association graph is built using large unlabeled social media corpus. It helps to derive contextual and grammatical analysis on both clean and noisy data.

As the future work we will extend our system to assure multilingual features. Also the grammatical features of the system are highly available for several improvements.

[1] J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, X. Deng, Exploiting topic based twitter sentiment for stock prediction, in: ACL (2), The Association for Computer Linguistics, 2013, pp. 24–29.

[2] C. Danescu-Niculescu-Mizil, M. Sudhof, D. Jurafsky, J. Leskovec, C. Potts, A computational approach to politeness with application to social factors, in: [23], 2013, pp. 250–259.

[3] T. Sakaki, M. Okazaki, Y. Matsuo, Earthquake shakes twitter users: Real-time event detection by social sensors, in: Proceedings of the 19th International Conference on World Wide Web, WWW '10, ACM, New York, NY, USA, 2010, pp. 851–860. URL: http://doi.acm.org/10.1145/1772690.1772777. doi:10.1145/1772690.1772777.

[4] A. Ritter, C. Cherry, B. Dolan, Unsupervised modeling of twitter conversations, in: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2010, pp. 172–180. URL: \url{http://scholar.google.de/scholar.bib?q=info:5hTIjtmFJKAJ:scholar.google.com/&output=citation&hl=de&as_sdt=0&ct=citation&cd=67}.

[5] B. Han, T. Baldwin, Lexical normalisation of short text messages: Makn sens a #twitter, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 368–378. URL: http://dl.acm.org/citation.cfm?id=2002472.2002520.

[6] H. Hassan, A. Menezes, Social text normalization using contextual graph random walks, in: [23], 2013, pp. 1577–1586.

[7] M. Choudhury, R. Saraf, V. Jain, A. Mukherjee, S. Sarkar, A. Basu, Investigation and modeling of the structure of texting language, Int. J. Doc. Anal. Recognit. 10 (2007) 157–174.

[8] E. Brill, R. C. Moore, An improved error model for noisy channel spelling correction, in: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, ACL '00, Association for Computational Linguistics, Stroudsburg, PA, USA, 2000, pp. 286–293. URL: http://dx.doi.org/10.3115/1075218.1075255. doi:10.3115/1075218.1075255.

[9] K. Toutanova, R. C. Moore, Pronunciation modeling for improved spelling correction, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, pp. 144–151. URL: http://dx.doi.org/10.3115/1073083.1073109. doi:10.3115/1073083.1073109.

[10] P. Cook, S. Stevenson, An unsupervised model for text message normalization, in: Proceedings of the Workshop on Computational Approaches to Linguistic Creativity, CALC '09, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, pp. 71–78. URL: http://dl.acm.org/citation.cfm?id=1642011.1642021.

[11] A. Aw, M. Zhang, J. Xiao, J. Su, A phrase-based statistical model for sms text normalization, in: Proceedings of the COLING/ACL on Main Conference Poster Sessions, COLING-ACL '06, Association for

Computational Linguistics, Stroudsburg, PA, USA, 2006, pp. 33–40. URL: http://dl.acm.org/citation.cfm?id=1273073.1273078.

[12] S. Gouws, D. Metzler, C. Cai, E. Hovy, Contextual bearing on linguistic variation in social media, in: Proceedings of the Workshop on Languages in Social Media, LSM '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 20–29. URL: http://dl.acm.org/citation.cfm?id=2021109.2021113.

[13] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, N. A. Smith, Improved part-of-speech tagging for online conversational text with word clusters, in: HLT-NAACL, The Association for Computational Linguistics, 2013, pp. 380–390.

[14] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, N. A. Smith, Part-of-speech tagging for twitter: Annotation, features, and experiments, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2, HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 42–47. URL: http://dl.acm.org/citation.cfm?id=2002736.2002747.

[15] V. Levenshtein, Binary Codes Capable of Correcting Deletions, Insertions and Reversals, Soviet Physics Doklady 10 (1966) 707.

[16] L. Philips, The double metaphone search algorithm, C/C++ Users J. 18 (2000) 38–43.

[17] D. Contractor, T. A. Faruquie, L. V. Subramaniam, Unsupervised cleansing of noisy text, in: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 189–196. URL: `http://dl.acm.org/citation.cfm?id=1944566.1944588`.

[18] M. Kaufmann, J. Kalita, Syntactic normalization of Twitter messages, in: Proceedings of the 8^{th} International Conference on Natural Language Processing (ICON 2010), Macmillan India, Chennai, India, 2010, pp. 149–158. URL: `http://ltrc.iiit.ac.in/icon\_archives/ICON2010/10Dec2010/Paper4-File33-Paper189.pdf`.

[19] J. Yang, J. Leskovec, Patterns of temporal variation in online media, in: I. King, W. Nejdl, H. Li (Eds.), WSDM, ACM, 2011, pp. 177–186.

[20] M. Lui, T. Baldwin, Langid.py: An off-the-shelf language identification tool, in: Proceedings of the ACL 2012 System Demonstrations, ACL '12, Association for Computational Linguistics, Stroudsburg, PA, USA, 2012, pp. 25–30. URL: `\url{http://dl.acm.org/citation.cfm?id=2390470.2390475}`.

[21] T. Baldwin, M. Lui, Language identification: The long and the short of the matter, in: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 229–237. URL: `\url{http://dl.acm.org/citation.cfm?id=1857999.1858026}`.

[22] F. Liu, F. Weng, X. Jiang, A broad-coverage normalization system for social media language, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Association for Computational Linguistics, 2012, pp. 1035–1044.

[23] conf/acl/2013-1, Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers, The Association for Computer Linguistics, 2013.