ICT Express
Information & Communications Technology

An approach to compress English sentences from Facebook posts using Machine Learning

Ankita Datta, Manaswita Datta, Puja Das, Subhankar Datta, Dwijen Rudrapal*

[1] Computer Sc & Engineering, National Institute of Technology, Agartala, India (email:ankitadatta.agt@gmail.com)
[2] Computer Sc & Engineering, National Institute of Technology, Agartala, India (email:dattamanaswita@gmail.com)
[3] Computer Sc & Engineering, National Institute of Technology, Agartala, India (email:puja250196@gmail.com)
[4] Computer Sc & Engineering, National Institute of Technology, Agartala, India (email:subhankardasnitacs@gmail.com)
[5] Computer Sc & Engineering, National Institute of Technology, Agartala, India (email:dwijen.rudrapal@gmail.com)
* Corresponding author

ABSTRACT
With the growing interest for automatic summarization of social media text like Facebook posts and tweets, a work on compression of such non-standard texts is in much need. Such sentence compression can be formally defined as the creation of an abbreviated sentence in a way that preserves the original meaning of the sentence and retains the grammaticality. In this paper, we propose an approach for compressing English sentences from Facebook posts by training the model on a parallel compression corpus to decide which word to delete or retain. We apply four machine learning algorithm (Sequential Minimal Optimization, Naive Bayes, Logistic Regression and Random Forest) on the corpus to evaluate the accuracy of the compressed sentences over a range of features based on lexical nature.

Index Terms: Sentence compression, Facebook posts, Machine Learning, Natural Language Processing

I. INTRODUCTION

Social media has received much attention over time. Now people are relying on social media like Facebook, Twitter for useful information about what is happening or what is being said about an entity. Social network sites have become the major source for automatic dissimilation of important information about events taking place all over the world. It provides the user with real-time news, information and public opinion about every subject matter. In short, this social media acts as the data powerhouse.

When needed, a user can search for a particular topic and retrieve the posts or tweets that contain the topic. But interpreting the information is quite challenging as there is no way the most significant information can be comprehended by the users from such a sheer volume of data. To make it easy for the users to find out the information they are interested in, an automatic summarization system is required that can generate a condensed content.

Although various approaches are proposed for summarization of social media text, those methods uses features such as term frequency [19], query relevance [20] and assign score to each sentence. Then extract the sentences which maximize the score to compose the intended summary. Those approaches don't shorten the sentence which leads to redundant words and phrases. Our work is motivated to solve this shortening problem.

Sentence compression is a subtask for text summarization. Besides automatic summarization of text, sentence compression is required for various other Natural Language Processing (NLP) processes like semantic role labeling, factoid questioning. The state-of-the-art sentence compression methods are only for standardized texts. Standardized texts are those found in newspaper articles, documents. But social media text differs from the former. The coarse nature of social media text creates certain difficulties for the existing methods [21]. Variation of writing style among users create a lot of challenges like misspelled words, non-standard abbreviation, typological errors, chat slangs etc. Detection of sentence boundary is difficult because of inconsistent punctuations and unwanted capitalization. Emoticons, letter repetition ("hiiii"), lack of grammaticality are other concerns for any syntactic analysis of such kind of text. Henceforth, the existing model fails to perform well on social media text.

According to Knight & Marcu [2], given an input source sentence of words $x = x_1, x_2, \ldots, x_n$ a compression is formed by dropping any subset of words. Good compressions are those which use fewer words than the source sentence, retain the most important information from the source sentence and remain grammatical. In this paper, we propose an approach to obtain the compressed version of a sentence by deletion of words or phrases. Initially, the corpus is divided into sentences. Each sentence is tokenized and tagged by a semi-automatic technique. A parallel corpus containing the compressed form of the source sentence is built for training. The experiment is evaluated on four machine learning algorithms. To the best of our knowledge, this is the first work on sentence compression for social media text.

The rest of the details are divided into the following sections: In section II, we discuss about the existing research that is relevant to sentence compression. Section

III describes the collection and annotation method used to built the corpus. The most interesting part of the paper is introduced in section IV, starting with the features used in the model, the evaluation metrics employed to measure the performance, followed by the accuracy of the four machine learning algorithm - Sequential Minimal Optimization (SMO), Naive Bayes (NB, Logistic Regression (LR) and Random Forest (RF). The performance analysis of the model is detailed in section V. Section VI provides direction for future work we plan to explore.

II. RELATED WORK

Several approaches for sentence compression of standardized text, both machine learning based and rule-based have been proposed to date, some of which are described below.

Some researchers [6, 7] used sentence compression to reduce redundancy in generating summaries while others used it to generate TV subtitles or text for small screens [1, 8]. Sentence compression is probably one of the most experimented text-to-text generation methods.

Some models formulated the task as a word or phrase deletion problem. For e.g., Knight & Marcu [2] proposed two models for sentence compression: a probabilistic noisy channel model and a decision-based model to generate abstractive summaries by dropping a subset of words. The deletion problem was further improvised to address global dependency in deleting local words [1, 8].

Also, some sophisticated models were introduced to improve the accuracy further. Two sentence reduction algorithms were proposed by Nguyen et al. [4]. The template-based learning model learns lexical translation rules and the Hidden Markov model learns which sequence of lexical rules should be applied to a sentence. The model performed better than Knight & Marcu [2] based on grammaticality and importance measure. Filippova & Strube [8] used rules for syntactic and semantic modification of parse tree to generate the compressed form. Cohn & Lapata [10] gave a compression method that extracts tree transduction rules from aligned, parsed texts. Then using a max-margin learning algorithm learns weights on transformations. Later, they [3] presented tree-to-tree transducer that is capable of transforming an input parse tree to a compressed parse tree based on Synchronous Tree Substitution Grammar. Lexicalized Markov Grammar was used by Galley & Mckeown [11] for sentence compression. Clarke & Lapata [12] used Integer Programming approach and encoded various decision variables and constraints in the model to guarantee the grammaticality of the reduced sentence.

The unsupervised model proposed by Cordeiro et al. [13] used Inductive Logic programming.(ILP). Banerjee et al. [9] used ILP for the generation of abstractive summaries by sentence compression. Most recently, Lu et al. [14] presented an approach based on a Re-read Mechanism and Bayesian combination model for sentences compression.

All of them are designed for standardized texts. Our work is conducted over social media texts or more precisely, Facebook posts.

III. DATA COLLECTION AND ANNOTATION

In this section, we describe the methodology adopted for collecting posts, selecting the required posts, tokenization of the corpus, preparing the parallel corpus and tagging the tokens.

A. Collecting posts

The first step was to collect a corpus of posts from Facebook. For this purpose, we used Facebook API [24] to fetch a total of 1884 posts related to UN election 2017 (CNN Facebook page [25]).

After a manual inspection, few posts were removed, being code-mixed or being non-English, as we restrict to English posts only. From these remaining posts, the first sentence of each was extracted since our task is related to sentence compression. Soon it was discovered that a proportion of the posts were identical. Those sentences were removed to make the posts unique in nature. Some sentences were also dropped because of the following reasons:

- Some sentences contain unwanted symbols or noise in an intermediate position.
- Some sentences didn't convey any meaning.
- Two sentences seemed like a single sentence because of lack of period or end marker.

To avoid this redundancy, 1200 unique and consistent sentences were manually selected for further processing.

B. Annotation Process

The collected data were annotated using a semi-automatic technique to speed up the manual tagging. Tokenization is difficult for social media texts because of the noisy and informal nature. Ritter's tagger [26] was employed to tokenize and tag the data in the first place. Although developed for tweets, it works even for Facebook posts.

While analyzing the corpus, it was found many sentences were not tokenized properly, as expected. This was mainly because of lack of white space between words (e.g. 1) or use of multiple periods between words (e.g. 2). Sometimes a word was tokenized into two because of use of punctuations (e.g. 3). All these complications were resolved manually.

1. In the sentence, "*Because he legitimately hasnt passed 1piece of legislation....*", '*1piece*' was treated as a single entity and was separated into two entities:'*1*', '*piece*'.
2. In "*Yeah liberals love a nationalist military state and a narcissistic leader with a bad haircut....oh wait!", 'haircut....oh*' was considered a single token. It was separated to form three tokens: '*haircut*', '....' and '*oh*'.

3. In the sentence, "*Leftists should like N Korea, Gov't runs everything*", '*Gov't*' was divided into two tokens: '*Gov'* and "t'. This was combined to form a single token.

From this 1200 sentences, a total of 16445 tokens are formed i.e. approximately 13.70 tokens for each sentence. The tagger also provided the POS tag, the chunk and the name entity for each token.

However, the accuracy of Ritter's tagger is only 88.3% [15]. To avoid erroneous tagging, manually inspection of the whole corpus was done by two annotators. For this purpose, the annotators were provided with the same file and instructed to manually check the POS tag and correct when errors are detected. In case of disagreement, they were asked to discuss until they reach a conclusion.

For preparing the parallel compression corpus, two annotator was provided with the same file. The instruction provided to the annotator is as follows:

The main aim of this task is sentence compression. You will be provided with a set of English sentences about the US election 2017 collected from Facebook. You are required to read each sentence and compress it to retain the grammaticality while keeping the most important information intact. The compressed sentence should be with the agreement of both of your consensus.

To do so, you are allowed to delete words or phrase from the source sentence. You are strictly restricted to insert, substitute or reorder the words of the sentence. You are not allowed to recreate two sentences from a single sentence when sentences are connected through conjunctions. You're also not allowed to delete any sentence from the original file. Sometimes you may come across sentences that cannot be shortened. In this case, keep the sentenced unedited.

Since there is no ideal compression form for sentences, all compressions will be considered valid as long as they retain the original meaning and are grammatically correct. Table 1 shows two examples of sentence compression.

Table 1. Two examples of sentence compression.

Source sentence	Compressed sentence
Hahahah Todd just contradicted his own argument	Todd contradicted his own argument.
Yeah liberals love a nationalist military state and a narcissistic leader with a bad haircut....oh wait!	liberals love a military state and a narcissistic leader!

The obtained parallel corpus is treated as the gold-standard dataset. It should be noted that all the annotators are familiar with the POS tagger through academic learning.

IV. EXPERIMENT

This section discusses the various rules from which the features are derived, the features used for training the classifier, the evaluation metrics adopted for the performance measure. It concludes with the result obtained from the experiment after applying four machine learning algorithms.

A. Rules

Since the parallel corpus is formed by deletion of a subset of words from a sentence, an observation is necessary to find the viable pattern by which this deletion was carried out. The following rules were observed:

- Delete adjective occurring to the left of a noun.
- Delete adverb occurring to the left of a verb.
- Delete the determiner.
- Name entity can be reduced to a single word (e.g. United States of America to America)

B. Features

Feature selection is the most important task of a machine learning approach. The features used in this model are based on the lexical and syntactic nature of the corpus and were formed after observing the rules. The features are mentioned and discussed, when required, below:

- The current word (W)
- The previous word (PW) and previous to previous word (PPW), next word (NW) and next to next word (NNW).
- The chunk (P): While analysis the parallel corpus, it was observed that either the whole chunk was removed or reduced to its corresponding head word to obtain the compressed sentence.
- POS tag of W, PW, PPW, NW, NNW, P: POS tagging is used to resolve the ambiguity that exists between words [5].
- Name-entity (NE): Name entities are nouns and hence kept intact for most of the sentences. These tokens are mostly not removed for compression.
- The head word of the chunk (H): The headword in a chunk is that word which is essential to the core meaning of the chunk [28]. For e.g. in case of a noun phrase, the head work is likely to be the subject. The head word of each chunk was identified by following Collins [23].

The former two lexical features are inspired from Jurafsky et al. [16] where the relationship existing between the current word and its surrounding words was examined.

C. Evaluation metrics

Often, accuracy is used to measure how correct the prediction of a classifier is. However since we have an imbalanced dataset, it will not be helpful. We, therefore use F-measure (harmonic mean of precision and recall) to evaluate the performance of our proposed model.

D. Result

A classifier is an algorithm that can predict the labels of unseen data. Choosing the right classifier is crucial for the performance of the model. For this experiment, a classifier that can detect whether a given word should be deleted or retained is needed. Since we are dealing with sentences extracted from Facebook, classifiers that work firmly for sequential tagging of data is required. The four classifiers used are namely Naive Bayes (NB), Logistic Regression (LR), Sequential Minimal Optimization (SMO) and Random Forest (RF).

A NB classifier is a binary classifier based on Bayes theorem and considers all the features to be independent. Empirical study shows that NB [18] is quite effective for sequential tagging of data. SMO is related to both Support Vector Machines (SVM) and optimization algorithms and performs well for linear SVM [22]. It is also popular for text classification, henceforth chosen. LR is selected because it performs well for correlated features. RF classifier averages multiple decision trees based on random samples from the dataset and proves to be effective for sequential data tagging.

On the dataset, 10-fold cross validation is performed i.e. 10% of the training data is given to the classifier. We use Weka [27] for implementing the machine learning algorithms. Table 2 describes the evaluation metric obtained after applying the algorithms on the dataset.

It can be clearly observed that SMO gives the highest weighted average F-measure (0.789) while LR (0.700) performed least. It is to be noted that the F-measure of LR and NB differ slightly. It should be noted that SMO and LR is evaluated on 5000 instances of the dataset.

Table 2. F-measure after 10-fold cross validation.

Evaluation metrics	SMO	NB	LR	RF
Weighted average F-score	0.789	0.770	0.700	0.759

V. DISCUSSION

Table 3 lists two compression examples resulted after compression. Although in most cases the grammaticality is not retained, still the intended meaning can be conveyed to some extent. Like for the first example, SMO resulted in a sentence almost like the manually compressed one. Further analysis showed that for most of the sentences, the length of the compressed version and original version didn't differ much.

Subjected to the the increased use of non-standard words, SMO gave better compression. Also, we carried out the task of tokenization semi-automatically. It definitely needs an automatic well-built mechanism for large corpus.

Table 3. F-measure after 10-fold cross validation

Source 1:	*i personally think that its disrespectful to the n korean people to go there.*
Compressed:	*its disrespectful to the n korean to go there.*
SMO:	*its disrespectful to the n korean people to go there.*
NB:	*that its disrespectful to the n korean people to go there.*
LR:	*think that its to the n korean people to gothere.*
RF:	*personally think its disrespectful to the n korean people to go there.*
Source 2:	*Normally i wouldn't be too happy about the govt telling me where i can and i cant go.*
Compressed:	*i wouldn't be happy about govt telling me where i can and i cant go.*
SMO:	*i wouldn't be too happy about the govt telling me where i can and i cant go.*
NB:	*i wouldn't be too happy about the govt telling me i can and i cant go.*
LR:	*i wouldn't be too happy about the telling i can and i cant go.*
RF:	*Normally wouldn't be too happy about the govt telling where i can and i cant go.*

VI. CONCLUSION AND FUTURE WORK

We present a machine learning approach to train social media text corpus for obtaining compressed texts. We detail on collecting, annotating the corpus using a semi-automatic technique and incorporate various features to determine whether a given word should be deleted or retained.

Drawing from the result obtained after applying four different machine learning algorithm (SMO, NB, LR, RF), SMO performed best with an F-measure of 0.789. Being the first attempt at social media text compression, we want to further modify the model to improve the grammaticality of the compressed sentence. We want to increase the size of the corpus substantially to decrease the number of unknown and non-standard words.

Owing to the extensive use of non-standard words, we want to employ lexical normalization [17]. Moreover, to evaluate the dataset more efficiently, we want to use Conditional Random Fields (CRF) algorithm.

ACKNOWLEDGMENTS

We would like to acknowledge our guide Mr. Dwijen Rudrapal, Assistant Professor, Computer Science and Engineering Department, National Institute of Technology, Agartala for his helpful suggestions.

REFERENCES

Journal Articles

[1] James Clarke., and Mirella Lapata, "Global inference for sentence compression: an Integer Linear programming approach," Journal of Artiicial Intelligence Research, vol. 31, pp. 399–429, Mar. 2008

[2] Kevin Knight and Daniel Marcu, "Summarization beyond sentence extraction: A probabilistic approach to sentence compression," Artificial Intelligence, vol. 139, pp. 91-107, July 2002

[3] Trevor Cohn and Mirella Lapata, "An abstractive approach to sentence compression," ACM Transactions on Intelligent Systems and Technology, vol. 4, Article 41, June 2013

[4] Minh Le Nguyen, Susumu Horiguchi, Akira Shimazu, and Bao Tu Ho, "Example-based sentence reduction using the hidden markov model," ACM Transactions on Asian Language Information Processing, vol. 3, pp. 146-158, June 2004

Books & Book Chapters

[5] Daniel Jurafsky & James H. Martin, "Part-of-Speech Tagging," in Speech and Language Processing, 3rd ed. draft, ch. 10, 2017, pp. 142-167.

Conference Proceedings

[6] Yuya Unno, Takashi Ninomiya, Yusuke Miyao, and Jun'ichi Tsujii, "Trimming CFG parse trees for sentence compression using machine learning approaches," in Proc. of the COLING/ACL on Main conference poster sessions, Sydney, Australia, 2006, pp. 850–857.

[7] Hongyan Jing, "Sentence reduction for automatic text summarization," in Proc. of the 6th conference on Applied Natural Language Processing, Stroudsburg, PA, USA, 2000, pp. 310–315.

[8] Katja Filippova and Michael Strube, "Dependency tree based sentence compression," in Proc. of the 5th International Natural Language Generation Conference, Stroudsburg, PA, USA, 2008, pp. 25-32.

[9] S. Banerjee, P. Mitra, and K. Sugiyama, "Multi-document abstractive summarization using ILP based multi-sentence compression," in Proc. of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 2015, pp. 1208-1214.

[10] Trevor Cohn and Mirella Lapata, "Sentence compression beyond word deletion," in Proc. of the 22nd International Conference on Computational Linguistics - vol. 1, Manchester, UK, 2008, pp. 137–144.

[11] Kathleen R. McKeown and Michel Galley, "Lexicalized Markov grammars for sentence compression," in Proc. of HLT Conference of the North American Chapter of the ACL, NY, USA, 2007, pp. 180–187.

[12] James Clarke and Mirella Lapata, "Constraint-based sentence compression: an Integer programming approach," in Proc. of the COLING/ACL on Main conference poster sessions, Sydney, Australia, 2006, pp. 144-151

[13] J Cordeiro, G. Dias, P. Brazdil, "Unsupervised induction of sentence compression rules," in Proc. of the Workshop on Language Generation and Summarisation, Singapore , 2009, pp. 15-22.

[14] Zhonglei Lu, Wenfen Liu, Yanfang Zhou, Xuexian Hu, Binyu Wang, "An effective approach of sentence compression based on re-read mechanism and Bayesian combination model," in Proc. of Chinese National Conference on Social Media Processing, Beijing, China, 2017, pp. 129-140

[15] Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith, "Improved part-of-speech tagging for online conversational text with word clusters," in Proc. of HLT Conference of the North American Chapter of the ACL, Atlanta, Georgia, 2013, pp. 380-390.

[16] Huihsin Tseng, Daniel Jurafsky, and Christopher Manning, "Morphological features help pos tagging of unknown words across language varieties," in Proc. of the 4th SIGHAN Workshop on Chinese Language Processing, Jeju Island, Korea, 2005.

[17] Tyler Baldwin and Yunyao Li, "An in-depth analysis of the effect of text normalization in social media," in Proc. of the North American Chapter of the ACL, Denver, Colorado, 2015, pp. 420–429.

[18] A. Najibullah, "Indonesian Text Summarization Based on Naïve Bayes Method," in Proc. of the International seminar and Conference: The Golden Triangle Interrelations in Religion, Science, Culture and Economics, Semarang, Indonesia, 2015.

[19] Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown, "A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization," in Proc. of the 29th Annual International SIGIR conference, Seattle, Washington, 2006, pp. 573-580.

[20] Surabhi Gupta, Ani Nenkova, and Dan Jurafsky, "Measuring importance and query relevance in topic-focused multi-document summarization," in Proc. of the 45th Annual Meeting of the ACL, Prague, Czech Republic, 2007, pp. 193-196.

[21] Jacob Einsenstein, "What to do about bad language on the internet," in Proc. of HLT Conference of the North American Chapter of the ACL, Atlanta, Georgia, 2013, pp. 359-369.

Technical reports

[22] J. Platt, 1997. "Sequential minimal optimization: a fast algorithm for training support vector machines". Technical Report MSR-TR-98-14, Microsoft Research.

Dissertations

[23] Michael Collins, 1999, "Head-driven statistical models for natural language parsing," Ph.D. thesis, University of Pennsylvania, Philadelphia.

Online Sources

[24] Facebook Inc. Facebook for developers. Available: https://developers.facebook.com/

[25] Facebook Inc. CNN Home page. Available: https://www.facebook.com/cnn/

[26] Alan Ritter. OSU Twitter NLP Tools. Available: https://github.com/aritter/twitter_nlp

[27] University of Waikato. Weka 3: Data mining software. Available: https://www.cs.waikato.ac.nz/ml/weka/

[28] University of Glasgow: Headword. Available: http://www.arts.gla.ac.uk/stella/lilt/headword.htm