Handling Normalization Issues for Part-of-Speech Tagging of Online Conversational Text

Géraldine Damnati[1], Jeremy Auguste[2], Alexis Nasr[2], Delphine Charlet[1]
Johannes Heinecke[1], Frédéric Béchet[2]
(1) Orange Labs, Lannion, France
{geraldine.damnati, delphine.charlet, johannes.heinecke}@orange.com
(2) Aix-Marseille Univ, Université de Toulon, CNRS, LIS
{jeremy.auguste, alexis.nasr, frederic.bechet}@univ-amu.fr

Abstract

For the purpose of POS tagging noisy user-generated text, should normalization be handled as a preliminary task or is it possible to handle misspelled words directly in the POS tagging model? We propose in this paper a combined approach where some errors are normalized before tagging, while a Gated Recurrent Unit deep neural network based tagger handles the remaining errors. Word embeddings are trained on a large corpus in order to address both normalization and POS tagging. Experiments are run on Contact Center chat conversations, a particular type of formal Computer Mediated Communication data.

Keywords: Part of Speech Tagging, Computer Mediated Communication, Spelling Error Correction

1. Introduction

Contact Center chat conversation is a particular type of noisy user generated text in the sense that it is a formal Computer Mediated Communication (CMC) interaction mode. It shares some normalization issues with other CMC texts such as chatroom conversations or social media interactions but unlike the aforementioned cases, the professional context implies some specificities. For instance, contact center logs are hardly prone to Internet slang. Another characteristic is that they are dyadic conversations with asymmetric levels of orthographic or grammatical errors. Agents may write with mistakes but are usually recruited for their linguistic skills, and can rely on predefined utterance libraries. Customers on the other hand can make mistakes for several different reasons, be it their educational background, linguistic skills, or even the importance they pay to the social perception of the errors they would make. Some of them will make no mistake at all while some others will misspell almost every word.

The purpose of this paper is to perform POS tagging on this particular type of Noisy User Generated text. Our goal is to study to which extent it is worth normalizing text before tagging it or directly handling language deviations in the design of the tagger. We will show that a good compromise is to handle some of the errors through lexical normalization but also to design a robust POS tagger that handles orthographic errors. We propose to use word embeddings at both levels: for text normalization and for POS tagging.

2. Related work

Text normalization has been studied for several years now, with different perspectives over time. When studying SMS style language, researchers tried to handle new phenomena including voluntary slang shortcuts through phonetic models of pronunciation (Toutanova and Moore, 2002; Kobus et al., 2008). Recently, the effort has been more particularly set on Social Media text normalization with specific challenges on Twitter texts (Baldwin et al., 2015), which has been shown to be more formal (Hu et al., 2013) that what is commonly expected. The typology of errors is slightly different and most recent works focus on one-to-one lexical errors (replacing one word by another). The availability of large corpora has led to the design of normalization lexicons (Han et al., 2012) that directly map correct words to there common ill-formed variants. (Sridhar, 2015) learns a normalization lexicon and converts it into a Finite State Transducer. More recently, the construction of normalization dictionaries using word embeddings on Twitter texts were performed for Brazilian Portuguese (Bertaglia and Nunes, 2016). In this paper, we focus on out-of-vocabulary words. We propose to generate variants of such words using a lexical corrector based on a customized edit distance and to use word embeddings as distributed representations of words to re-rank these hypotheses thanks to contextual distance estimation.

In order to adapt POS tagging systems for noisy text, several approaches have proposed to use word clusters provided by hierarchical clustering approaches such as the Brown algorithm. (Owoputi et al., 2013) use word clusters along with dedicated lexical features to enrich their tagger in the context of online conversations. (Derczynski et al., 2013) use clustering approaches to handle linguistic noise, and train their system from a mixture of hand-annotated tweets and existing POS-labeled data. (Nasr et al., 2016) address the issue of training data mismatch in the context of online conversations and show that equivalent performance can be obtained by training on a small in domain corpus rather than using generic POS-labeled resources.

3. Text normalization

Our text normalization process operates in two steps, the first one produces in-lexicon variants for an out of lexicon form. The second one reranks the forms produced by the first step, using a distributional distance. The first step is based on a lexicon and an edit distance while the second relies on word embeddings. We focus on one-to-one normalization, avoiding the issue of agglutinations or split words.

3.1. Defining a target lexicon

In order to generate correct hypotheses of an out of vocabulary form, we need to define a target lexicon. A lexicon should both reflect general language common terms and company related specific terms. If the general common terms lexicon is very large, a lexical corrector would have more chances to propose irrelevant out of domain alternatives. Hence, we have chosen to reduce the size of our lexicon by selecting words that appear more than 500 times in the French Wikipedia, resulting in 36,420 words. Additionally a set of 388 manually crafted domain specific terms was added to the lexicon. The latter were obtained by selecting words in the manually corrected training corpus that were not covered by the general lexicon. Finally, as case is not a reliable information in such data we reduce all words of the lexicon to their lower case form. Contrastive experiments have been run but are not reported in this extended absract, showing that the choice of the lexicon is important for the whole process. Including a general knowledge lexicon from Wikipedia is more helpful for correcting Agent errors than for correcting Customer errors.

3.2. Edit-distance based normalization

The corrector built with this lexicon is based on the Damerau-Levenshtein (DL) distance. The code of the lexical corrector is available at the above mentioned url [1]. In contrast to standard DL we assign weights to error types: missing or superfluous diacritics only add 0.3 to the distance. Additionally, adjacent letters on the keyboard (like an `e` instead of an `r`, which sits just next to each other on QWERTY and AZERTY keyboards), add 0.9 to the edit-distance. Letter transpositions (such as `teh` instead of `the`) also account for 0.9. All other differences account for 1 in the global distance. These weights are configurable and have been optimized for our task.

Words to be processed are all transformed to their lower case form before applying the corrector with the lower case lexicon described in 3.1. The original case is reintroduced before applying the POS tagger.

The lexical corrector provides a list of candidates for correction, until a maximum cost is reach. This upper bound is proportional to the word length n in terms of number of letters and is computed as follows: $max\_cost = n \times \gamma$

In these experiments γ is set to 0.3. Here again contrastive experiments can be provided showing the impact of the γ parameter.

As we are dealing with formal interactions,we did not apply the modification on the edit distance proposed by (Hassan and Menezes, 2013) where edit distance is computed on consonant skeletons, nor do we use Longest Common Subsequence Ratio (LCSR) as it didn't reveal to be helpful in our case.

3.3. Rescoring with word embeddings

The edit distance based variant generation process described above does not take into account the context of a word when generating variants. In order to take it into account, we propose to rescore the hypothesized alternatives using a distance metric derived from the cosine similarity between word embeddings. We have gathered a large amount of unannotated chat conversations from the same technical assistance domain, resulting in a 16.2M words corpus, denoted BIG. For the particular purpose of lexical normalization we are more interested in paradigmatic associations than in syntagmatic associations. Hence `word2vec` is used with a small window size of 4. Furthermore, in order to capture as many tokens as possible we have chosen to keep all tokens occurring at least twice in the corpus when learning the word embeddings. The lexicon produced contains 43.4K forms.

Let w be an observed form and $\alpha_i(w)$ be the i^{th} alternative proposed by the edit distance based lexical corrector. Let V_{emb} be the vocabulary of the word vector model estimated on the large unannotated corpus, and v_w denote the vector of word w. The word embeddings based distance $d_{emb}(w, \alpha_i(w))$ is defined as $1 - \cos(v_w, v_{\alpha_i(w)})$. If either v or $\alpha_i(w)$ does not belong to V_{emb}, $d_{emb}(w, \alpha_i(w))$ is set to 1, meaning that it will not have any effect on the re-scoring process. Let $C(w, \alpha_i(w))$ be the edit cost provided by the lexical corrector between w and the proposed alternative $\alpha_i(w)$, the rescoring process simply consists in multiplying the edit score by the distance derived from the embeddings.

$$C_{emb}(w, \alpha_i(w)) = C(w, \alpha_i(w)) \times d_{emb}(w, \alpha_i(w))$$

4. Part of speech tagging

The part of speech tagger used in our experiment is based on Gated Recurrent Units (GRU). GRUs, introduced by (Cho et al., 2014), are recurrent neural networks that work in a similar fashion than LSTMs. GRUs are simpler than LSTMs: they do not have an output gate, and the input and forget gates are merged into an update gate. This property allows GRUs to be computationally more efficient.

The ability of GRUs to handle long distance dependencies make them suitable for sequence labeling tasks, such as POS tagging. Our tagger uses a bidirectional GRU making use of past and future features for each specific word in a sentence. The bidirectionnal GRU consists of a forward layer and a backward layer which outputs are concatenated. The forward layer processes the sequence from the start to the end, while the backward layer processes it from the end to the start.

The input of the network is a sequence of words with their associated morphological and lexical features. The words are encoded using a lookup table which associates each word with its word embedding representation. These word embeddings can be initialized with pretrained embeddings and/or learned when training the model. For the morphological and typographic features, we use a boolean value for the presence of an uppercase character as the first letter of the word as well as the word suffixes of length 3 and 4 represented as onehot vectors. Finally, we also input as onehot vectors external lexicon information, constructed using the *Lefff* lexicon (Sagot, 2010). Such vectors represent the possible part-of-speech labels of a word. On the output layer, we use a softmax activation. During training, categorical

[1] https://github.com/Orange-OpenSource/lexical-corrector

cross-entropy is used as the loss function and the Adam optimiser (Kingma and Ba, 2014) is used for the gradient descent optimisation.

5. Experiments and results

The corpus used for our experiments has been extracted from chat conversation logs of a French technical assistance contact center. A set of 91 conversations has been fully manually corrected and POS tagged. This corpus has been split in two equal parts: the TRAIN part being used to train the POS tagger and the TEST part for evaluation. Both sets contain around 17K words, with 5.4K words from the Customer side 11.6K words from the Agent side.

The typology of errors follow the one proposed in (Nasr et al., 2016). `DIACR` stands for *diacritic* errors which are common in French, `APOST` for missing or misplaced *apostrophe*, `AGGLU` for *agglutinations* and `SPLIT` for words split into two words. It is common in French to find confusions `INFPP` between past participles and infinitives for verbs ending with `er` (`j'ai changé` ↔ `j'ai changer`). Morpho-syntactic inflection `INFL` in French is error prone as it is common that different inflected forms of a same word are homophones. `MOD1C` correspond to one modified character (substituted, deleted or inserted) or when two adjacent letters are switched.

5.1. Text Normalization Evaluation

In Table 1, we present the results of the text normalization steps, on the whole corpus. $editonly$ refers to text processed by the edit-based correction. $editembed$ refers to the full correction process with semantic rescoring based on word embedding distances. We show in the first line the amount of word errors that are potentially correctable by the proposed approach (*i.e.* errors leading to Out-of-Vocabulary words) and the remaining subset of word errors which can not be corrected by our approach (errors resulting in in-vocabulary words and words discarded from the correction process). Among the total amount of 1646 erroneous words, 53% (870) are potentially correctable. The other 47% are words that do appear in our lexicon. After the edit-based correction step, 76.7% of these errors have been corrected, leading to 202 remaining errors. When rescoring with semantic similarity, the $editembed$ approach enables to correct 5% additional words, leading to an overall correction of 81.6% of the potentially correctable errors. It is worth noticing that in our approach, as the lexicon used in the correction step is not exhaustive, we observe 80 added errors due to the fact that some words, which were correct in the raw text, but not present in the lexicon, have been erroneously modified into an in-vocabulary form. Overall, the word error rate (WER) on raw text was 4.37% and is reduced to 2.81% after $editonly$, and to 2.7% after $editembed$ semantic rescoring. When restricting the corpus to the Customer messages, the initial WER reaches 9.82% and the normalization process leads to 5.07%.

Detailed error numbers according to the type of errors, on TEST only, can be found in Table 3. As expected, the proposed approach is efficient for diacritics, apostrophes and 1 letter modifications (`DIACR, APOST, MOD1C`). However it is inefficient for agglutination `AGGLU` and `SPLIT`, for confusion of verbal homophonic forms (`INFPP`) and for inflexion errors (`INFL`).

	raw	$editonly$	$editembed$
# of correctable err.	870	202	160
# of non correctable err.	776	776	776
overall WER	4.37	2.81	2.70
CUST. WER	9.82	5.35	5.07
AGENT WER	1.73	1.58	1.55

Table 1: Evaluation of normalization. Number of correctable and non-correctable errors (second and third lines) and word error rates (lines four to six). Third and fourth columns indicate the errors after normalizing the text with respectively the edit distance based and the word embedding distance based normalization.

5.2. Part of speech tagging results

Three different taggers have been trained on the corrected version of the train corpus[2]. They differ in the embeddings that were used to represent the words. The first tagger does not use any pre-trained embeddings, the second uses embeddings trained on the raw corpus while the third one uses embeddings trained on the automatically corrected corpus. Three versions of the test corpus have been taken as input to evaluate the taggers. The raw version, the gold version, which has been manually corrected and the auto version, which has been automatically corrected. The accuracy of the three taggers on the three versions of the test corpus are represented in Table 2. POS accuracy has been computed on the whole TEST corpus as well as on subsets of the TEST corpus produced by the agents and the customers.

input	ALL	AGENT	CUST.
no pretrained embeddings			
gold	95.37	96.39	93.39
auto	93.83	95.52	90.54
raw	93.07	95.31	88.70
embeddings trained on raw corpus			
gold	95.36	96.37	93.40
auto	94.25	95.78	91.25
raw	94.01	95.77	90.60
embeddings trained on corrected corpus			
gold	95.35	96.35	93.42
auto	94.13	95.62	91.24
raw	93.43	95.52	89.37

Table 2: POS tagging accuracy of the three taggers on the test corpus. For each tagger results are given on three versions of the corpus: manually corrected (gold), automatically corrected (auto) and raw. The two last columns indicate accuracy on the Agent and Customer parts of the corpus.

Table 2 shows that the three taggers reach almost the same performances on the gold version of the TEST corpus. The

[2]Taggers trained on the raw versions of the corpus yielded lower results.

best performances on the raw TEST corpus are obtained by the second tagger, which word embeddings have been trained on the raw BIG corpus. This result does not come as a surprise since the raw TEST corpus contains spelling errors that could have occurred in the raw BIG corpus and therefore have word embedding representations. Although the tagger that uses pretrained word embeddings yields better results than the first tagger, it is still beneficial to automatically correct the input prior to tagging. Table 2 also shows that the benefits of using word embeddings trained on the raw BIG corpus is higher on the customer side, which was also expected since this part of the corpus contains more errors. Using embeddings trained on the automatically corrected BIG corpus doesn't yield any further improvements, suggesting that the initial embeddings trained on the raw corpus already capture the relevant information.

The influence of the spelling errors on the tagging process is analysed in Table 3. Each line of the table corresponds to one type T of spelling error. The left part of the table presents the results on the raw version of the test data and the right part, on its corrected version. The first column in each part is the total number of occurrences of type T errors, the second column is the number of type T errors that also correspond to a tagging error and the third column, the part of T errors that correspond to a tagging error (ratio of columns 1 and 2). The tagger used here is the second one, which uses word embeddings trained on raw data. Table 3 shows that the type of spelling error that is the more POS error prone is the `INFPP` type, which almost always lead to a tagging error. More generally, the table shows that the correction process tends to correct errors that are not very harmful to the tagger. This is especially true for the diacritic errors: the correction process corrects 67% of them but the number of tagging errors on this type of spelling errors is only decreased by 32.3%. Actually the remaining diacritic errors are typically errors on frequent function words in French that have different categories (`où` $\leftrightarrow$ `ou`, `à` $\leftrightarrow$ `a` meaning `where` $\leftrightarrow$ `or`, `to` $\leftrightarrow$ `have`).

	raw			auto		
ERR Type	Spell	Tag	ratio	Spell	Tag	ratio
DIACR	250	96	38.40	81	65	80.25
APOST	47	5	10.64	11	3	27.27
MOD1C	135	44	32.59	77	26	33.77
AGGLU	57	47	82.46	54	46	85.19
SPLIT	31	24	77.42	31	24	77.42
INFPP	29	26	89.66	29	26	89.66
INFL	84	9	10.71	77	8	10.39
OTHER	50	20	40.00	40	22	55.00

Table 3: POS tagging errors with respect to spelling error types, on raw and on corrected input. Column *Spell* is the number of errors of the corresponding type, Column *Tag* is the number of errors of the type that also correspond to tagging errors, column *ratio* is the ratio of column *Tag* and *Spell*.

6. Conclusion

We have shown in this paper that word embeddings trained on a noisy corpus can help for both tasks of correcting misspelled words and POS tagging noisy input. We have also quantified the impact of spelling errors of different categories on the POS tagging task. We plan as future work to combine both processes in a single one that performs both POS tagging and correction.

Acknowledgements

This work was partially funded by the French Agence Nationale pour la Recherche (ANR) under the grant ANR-15-CE23-0003

7. Bibliographical References

Baldwin, T., De Marneffe, M. C., Han, B., Kim, Y.-B., Ritter, A., and Xu, W. (2015). Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text (WNUT 2015), Beijing, China*.

Bertaglia, T. F. C. and Nunes, M. d. G. V. (2016). Exploring word embeddings for unsupervised textual user-generated content normalization. *Proceedings of the Workshop on Noisy User-generated Text (WNUT 2016), Osaka, Japan*.

Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Derczynski, L., Ritter, A., Clark, S., and Bontcheva, K. (2013). Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *RANLP*, pages 198–206.

Han, B., Cook, P., and Baldwin, T. (2012). Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 421–432. Association for Computational Linguistics.

Hassan, H. and Menezes, A. (2013). Social text normalization using contextual graph random walks. In *ACL (1)*, pages 1577–1586.

Hu, Y., Talamadupula, K., Kambhampati, S., et al. (2013). Dude, srsly?: The surprisingly formal nature of twitter's language. In *Proceedings of the International AAAI Conference on Web and Social Media*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Kobus, C., Yvon, F., and Damnati, G. (2008). Normalizing sms: are two metaphors better than one? In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 441–448. Association for Computational Linguistics.

Nasr, A., Damnati, G., Guerraz, A., and Bechet, F. (2016). Syntactic parsing of chat language in contact center conversation corpus. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 175.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT*, pages 380–390.

Sagot, B. (2010). The lefff, a freely available and large-coverage morphological and syntactic lexicon for french. In *7th international conference on Language Resources and Evaluation (LREC 2010)*.

Sridhar, V. K. R. (2015). Unsupervised text normalization using distributed representations of words and phrases. In *Proceedings of NAACL-HLT*, pages 8–16.

Toutanova, K. and Moore, R. C. (2002). Pronunciation modeling for improved spelling correction. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 144–151. Association for Computational Linguistics.