CoNLL 2014 Shared Task: Grammatical Error Correction with a Syntactic N-gram Language Model from a Big Corpora

S. David Hernandez
Centro de Investigación en
Computación - IPN / México
shernandez_b12@sagitario.cic.ipn.mx

Hiram Calvo
Centro de Investigación en
Computación - IPN / México
hcalvo@cic.ipn.mx

Abstract

We describe our approach to grammatical error correction presented in the CoNLL Shared Task 2014. Our work is focused on error detection in sentences with a language model based on syntactic tri-grams and bi-grams extracted from dependency trees generated from 90% of the English Wikipedia. Also, we add a naïve module to error correction that outputs a set of possible answers, those sentences are scored using a syntactic n-gram language model. The sentence with the best score is the final suggestion of the system.

The system was ranked 11th, evidently this is a very simple approach, but since the beginning our main goal was to test the syntactic n-gram language model with a big corpus to future comparison.

1 Introduction

Grammatical error correction is a difficult task to solve even for humans, because there are a lot of phenomena that can occur in a sentence. One example of the difficulty of the task is that the annotators of the training and test data in the NUCLE (Dahlmeier et al., 2013) differs in the corrections that they made to the sentences, those differences in the annotations are mostly because depend on uncontrolled conditions, such knowledge, emotional state and the environment of the annotator at the moment that the task is performed. This time the shared task is more difficult than the last year (Ng and Wu et al., 2013) that considered only five types of errors, and this time the task consist into correct all the grammatical errors in the NUCLE (Ng and Wu et al., 2014).

We are interested into test the behaviour of different methods used in different NLP task with the syntactic n-grams as a resource, in order to set a baseline to future work. There is work that probes that there is an improvement using syntactic n-grams in (Sidorov and Velasquez et al., 2014) where the author uses syntactic n-grams as machine learning features, another example of the use of syntactic n-grams occurred in the CoNLL 2013 Shared Task in (Sidorov and Gupta et al., CoNLL 2013), but they used a different approach from us.

Until the moment we do not have a comparison with the same method that we used in this task using normal n-grams, still our hypothesis is that syntactic n-grams allow us to relate words that in a common n-gram model wouldn't be related and that can outperform the results.

For example, in the sentence:

"Genetic risk refers more to your chance of inheriting a disorder or disease ."

Some common tri-grams are *"to your chance"*, *"your chance of"*, *"chance of inheriting"*. The word *chance* can not be related to the words *"disorder"* or *"disease"*, unless we use 5-grams or 7-grams, unlike with the syntactic tri-grams that as can be appreciated in the Table 3 the relation between this words are normally included.

Another hypothesis is that a low probability in a syntactic n-gram is an indicator that exist a wrong token in the portion of a dependency tree. A simple example of this intuition can be seen in the Table 1 for the sentence "This will , if not already , caused problems as there are very limited spaces for us ." from the training data in the NUCLE. The bold words are wrong tokens annotated in the training data and the numbers are the token number in the sentence.

As can be observed the low probability syntactic tri-grams include the wrong tokens. The problem is to establish a threshold in the prob-

Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, pages 53–59,
Baltimore, Maryland, 26-27 July 2014.

q_i	Syntactic tri-grams
0.0	'**are-12 spaces-15** us-17 True'
0.0	'**spaces-15** limited-14 us-17 False'
0.00004	'**caused-8** will-2 **are-12** False'
0.00004	'**caused-8** will-2 not-5 False'
0.00004	'**caused-8** will-2 This-1 True'
0.00004	'**caused-8** will-2 problems-9 False'
0.00029	'**caused-8** not-5 **are-12** False'
0.00047	'**caused-8 are-12** as-10 True'
0.00054	'**are-12 spaces-15** limited-14 True'
0.00054	'**caused-8 are-12 spaces-15** True'
0.00057	'**caused-8 are-12** there-11 True'
0.00065	'**caused-8** problems-9 **are-12** False'
0.00109	'**spaces-15** limited-14 very-13 True'
0.00194	'**caused-8** not-5 already-6 True'
0.00314	'**caused-8** not-5 problems-9 False'
0.00522	'**caused-8** not-5 if-4 True'
0.22841	'**are-12** as-10 there-11 False'
0.375	'**are-12** as-10 **spaces-15** False'
0.75510	'**are-12** there-11 **spaces-15** False'
1.0	'ROOT-0 **caused-8 are-12** True'
1.0	'ROOT-0 **caused-8** not-5 True'
1.0	'ROOT-0 **caused-8** problems-9 True'
1.0	'ROOT-0 **caused-8** will-2 True'
1.0	'not-5 if-4 already-6 False'

Table 1: Ordered probabilities of syntactic trigrams. The wrong tokens are *"caused"*, *"are"* and *"spaces"*.

abilities to consider as wrong a syntactic trigram and separate the wrong tokens from the correct ones.

2 Resources

For the language model we used a Wikipedia dump as training data (Wikipedia, 2013) and extracted the text with the Multithread Wikipedia Extractor (Souza, 2012) then was tokenized with the Stanford Tokenizer (Manning et al.,). There are about 87 millions of sentences and 1,480 millions of tokens.

To generate the dependency trees we used the Stanford Parser 3.2 (Socher et al., 2013), but for the syntactic n-gram language model we only took 90% of the sentences randomly chosen. The parsing task took a lot of time to be made with our computing resources and we had to use threads with the Stanford Parser, unfortunately this increases the amount of memory required by the software, so we had to exclude the sentences with more than one hundred token. At the end we parsed about 75 millions of sentences.

The dependency trees were generated as Stanford typed dependencies (Marneffe et al., 2006), in specific in the collapsed with propagation version as described in (Marneffe et al., 2008). One example of this kind of dependencies can be seen in the Figure 2. As can be observed the collapsed with propagation typed dependencies can break the tree, so strictly this is a directed graph with the grammatical relations in the edges and the words of the sentence in the nodes, though as convention we will continue referring it as a tree. In total there are about 1,166 million grammatical relations.

In the error detection phase we used the information provided with the NUCLE (Dahlmeier et al., 2013), specifically the tokens, POS and the grammatical relations from the test data in CoNLL style. From the training data we only made some calculations about the kinds of errors that occur with higher frequency and we used this information to include some rules in the correction phase.

3 System description

3.1 Syntactic n-gram language model

We used the dependency trees from Wikipedia corpus to generate the syntactic n-grams in the non-continuous form as described in (Sidorov, 2013) and in the book (Sidorov, Book 2013), but there is an significant difference, the current work with syntactic n-grams was made with the basic dependencies, and as we said before, we are using the dependencies that collapses the prepositions and propagates the conjunctions. The tree in Figure 1 is in the Basic representation and the differences with the collapsed and propagated dependencies can be appreciated in the Figure 2.

This change allow us to increase the scope of the relations between content words, but also it makes difficult to find preposition errors, so our system do not consider preposition correction.

The tables 2 and 3 show the syntactic trigrams generated whit each one of the dependency representations, but without the relations for lack of space. As can be observed the

Genetic risk refers more to your chance of inheriting a disorder or disease

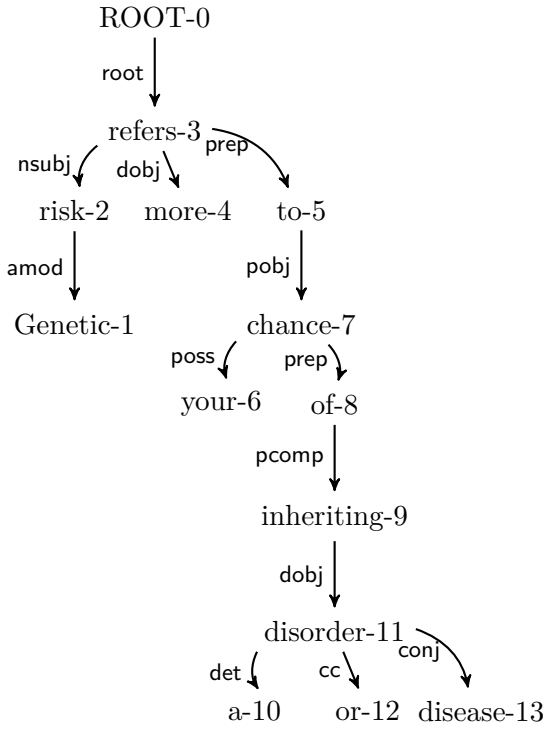

Figure 1: Basic dependencies.

Genetic risk refers more to your chance of inheriting a disorder or disease

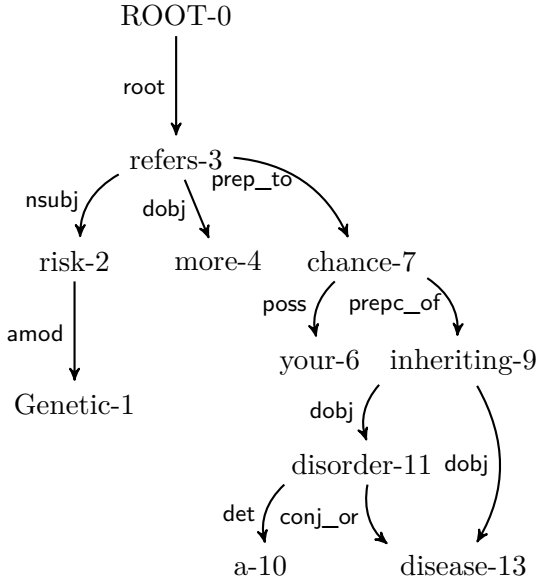

Figure 2: Collapsed dependencies with propagation.

word *"chance"* in the basic dependencies is not directly related with the words *"disorder"* and *"disease"*, on the contrary with the collapsed and propagated dependencies.

w_1	w_2	w_3	Continuous
to-5	chance-7	of-8	True
to-5	chance-7	your-6	True
refers-3	to-5	chance-7	True
refers-3	risk-2	Genetic-1	True
of-8	inheriting-9	disorder-11	True
inheriting-9	disorder-11	a-10	True
inheriting-9	disorder-11	disease-13	True
inheriting-9	disorder-11	or-12	True
chance-7	**of-8**	**inheriting-9**	True
ROOT-0	refers-3	to-5	True
ROOT-0	refers-3	risk-2	True
ROOT-0	refers-3	more-4	True
refers-3	risk-2	to-5	False
refers-3	risk-2	more-4	False
refers-3	more-4	to-5	False
disorder-11	a-10	disease-13	False
disorder-11	a-10	or-12	False
disorder-11	or-12	disease-13	False
chance-7	your-6	of-8	False

Table 2: Syntactic tri-grams from the basic dependencies.

w_1	w_2	w_3	Continuous
refers-3	chance-7	inheriting-9	True
refers-3	chance-7	your-6	True
refers-3	risk-2	Genetic-1	True
inheriting-9	disorder-11	a-10	True
inheriting-9	disorder-11	disease-13	True
chance-7	**inheriting-9**	**disorder-11**	True
chance-7	**inheriting-9**	**disease-13**	True
ROOT-0	refers-3	chance-7	True
ROOT-0	refers-3	risk-2	True
ROOT-0	refers-3	more-4	True
refers-3	risk-2	chance-7	False
refers-3	risk-2	more-4	False
refers-3	more-4	chance-7	False
inheriting-9	disorder-11	disease-13	False
disorder-11	a-10	disease-13	False
chance-7	your-6	inheriting-9	False

Table 3: Syntactic tri-grams from the collapsed with propagation dependencies.

Next we show the maximum likelihood estimations that we calculated for this language model. Where $w_1, w_2, w_3 \in W$ and W is the set of words of the sentence, $r_1, r_2 \in R$ with R as the set of grammatical relations between the words and $c \in \{True, False\}$, with True representing a continuous syntactic n-gram and False a non-continuous syntactic n-gram.

In equation (1) we take the maximum value between the probability estimation of the tri-gram with and without grammatical relations in order to favour the complete tri-gram.

Even with a big corpus as Wikipedia and with the non-continuous syntactic tri-grams these estimations can produce zeros in the probabilities, then we have to draw upon a back-off, so, we add other estimations.

$$q_1 = max(q(w_1|w_2, w_3; r_1, r_2; c), \\ q(w_1|w_2, w_3; c)) \quad (1)$$

$$q_2 = max(q(w_3|w_1, w_2; r_1, r_2; c), \\ q(w_3|w_1, w_2; c)) \quad (2)$$

Notice that equation (2) is similar to (1), both evaluate the same syntactic tri-gram, but with a different word of interest.

$$q_3 = \begin{cases} min(q(w_2|w_1; r_1), q(w_3|w_2; r_2)) & if \quad c = True \\ min(q(w_2|w_1; r_1), q(w_3|w_1; r_2)) & if \quad c = False \end{cases} \quad (3)$$

$$q_4 = \begin{cases} min(q(w_2|w_1), q(w_3|w_2)) & if \quad c = True \\ min(q(w_2|w_1), q(w_3|w_1)) & if \quad c = False \end{cases} \quad (4)$$

$$q_5 = max(q_3, q_4) \quad (5)$$

$$q_6 = \begin{cases} min(q(w_1|w_2; r_1), q(w_2|w_3; r_2)) & if \quad c = True \\ min(q(w_1|w_2; r_1), q(w_1|w_3; r_2)) & if \quad c = False \end{cases} \quad (6)$$

$$q_7 = \begin{cases} min(q(w_1|w_2), q(w_2|w_3)) & if \quad c = True \\ min(q(w_1|w_2), q(w_1|w_3)) & if \quad c = False \end{cases} \quad (7)$$

$$q_8 = max(q_6, q_7) \quad (8)$$

When the probabilities in equations (1) and (2) are equal to zero, we add a back-off estimation based in syntactic bi-grams, since a syntactic tri-gram is formed of two syntactic bi-grams or grammatical relations with different probabilities, but both or one of them can contain wrong tokens, so we decided to penalize the complete probability estimation of the syntactic tri-gram by choosing the min probability between the two relations. In the equations (3), (4), (6) and (7) a min operation is included to penalize the low probability in a syntactic bi-gram that corresponds to a syntactic tri-gram. In the equations (5) and (8) the max operation plays the same role as in equations (1) and (2).

The final expression of the model is shown in equation (9).

$$q_{stri} = \begin{cases} q_1 & if \quad q_1 > 0 \\ q_2 & if \quad q_1 = 0 \quad and \quad q_2 > 0 \\ q_5 & if \quad q_2 = 0 \quad and \quad q_5 > 0 \\ q_8 & if \quad q_5 = 0 \quad and \quad q_8 > 0 \\ 0 & Otherwise \end{cases} \quad (9)$$

Where $q_{stri} = q(w_1, w_2, w_3; r_1, r_2; c)$ and represents the probability of the syntactic tri-gram.

The syntactic tri-grams continuous and non-continuous produced a vast amount of data, for that reason we only took about 1,660 millions of syntactic tri-grams to made the language model. This data can be downloaded from (Syntactic N-grams, 2014).

3.2 Detection and correction

In order to detect errors in the test data of NUCLE (Dahlmeier et al., 2013), we extract the Stanford typed dependencies from the conll-style file and to be congruent with the data of our language model excluded the *punct* grammatical relations. Then we obtain the syntactic tri-grams and probabilities of each sentence. The assumption is that low probability in a syntactic tri-gram makes it a candidate to be wrong, since grammatical errors could produce trees with portions where grammatical relations are unseen in the training data or with a low probability.

q_i	Syntactic tri-grams
0.0	refers-3 more-4 chance-7 False
0.0	refers-3 risk-2 chance-7 False
0.0	refers-3 chance-7 your-6 True
0.0	refers-3 chance-7 inheriting-9 True
0.00015	refers-3 risk-2 Genetic-1 True
0.00023	refers-3 risk-2 more-4 False
0.00355	chance-7 your-6 inheriting-9 False
0.00355	chance-7 inheriting-9 disorder-11 True
0.00609	inheriting-9 disorder-11 disease-13 True
0.00609	inheriting-9 disorder-11 a-10 True
0.00609	inheriting-9 disorder-11 disease-13 False
0.02128	disorder-11 a-10 disease-13 False
1.0	ROOT-0 refers-3 more-4 True
1.0	ROOT-0 refers-3 risk-2 True
1.0	ROOT-0 refers-3 chance-7 True
1.0	chance-7 inheriting-9 disease-13 True

Table 4: Ordered probabilities of the syntactic tri-grams.

To add the wrong syntactic tri-grams to a set E we include two parameters, $\alpha = 0.0001$

which is a threshold and $\xi = 0.5$ that is a percentage. To decide whose syntactic trigrams must be in the set E, we sort them upwardly as in the table 4, if satisfy the condition $(q_i < \alpha)$ and $(q_i >= \xi q_{i+1})$ for $i \in \{1, 2, ...,$ until the first exception $\}$ the syntactic tri-gram is added to the set E. The fixed values of α and ξ were selected by experimentation.

w_1	w_2	w_3	Continuous
refers-3	more-4	chance-7	False
refers-3	risk-2	chance-7	False
refers-3	chance-7	your-6	True
refers-3	chance-7	inheriting-9	True

Table 5: Set of possible wrong syntactic trigrams.

The syntactic tri-grams in the table 5 are the selected as suspicious to be wrong with the above considerations. All the tokens can be part of a grammatical error, but to get replacement candidates of all of them can increase the complexity of the task and with the window of time that we had to accomplish the task, so we decided to select words in the set E to be considered as wrong tokens. We counted the total amount of occurrences of each token in the set E and took the two with higher values.

Count	Tokens
4	refers-3
4	chance-7
1	more-4
1	risk-2
1	your-6
1	inheriting-9

Table 6: Possible wrong tokens.

We chose the best candidates that can replace each word in the sentence and generate new sentences with each one of the candidates in his different combinations. Is easy to see that can be a lot of sentences, considering that each word can have more than one candidate and that each sentence could have more than one wrong token to be replaced. To obtain the candidates to each suspicious token we search in our training data, words that start with the stemmed form of the selected token and that depends of the same word with the same relation, also we add the lemmatized word. The lemmatization was made with the WordNetLemmatizer and the stemming with LancasterStemmer, both from NLTK. Also we applied as we said some naïve rules based on the most frequent errors in the training corpus from NUCLE, for example, when the suspicious token is a pronoun or a common verb as *"have"* or *"do"* we replace them with their conjugations.

For the example in table 6, we have the respective candidates in table 7. Visibly the word *"chants"* has nothing to do with the original token to be replaced, it shows the main reason of why we have low score, the rules used in the correction phase are very simple. For this example, the word *"chance"* stemmed with the LancasterStemmer is *"chant"*, then the search of words in the grammatical relations that depends on the word *"refers"* and with the same relation, outputs the word *"chants"*.

Tokens	Candidates
refers	refers
chance-7	chance, chants

Table 7: Candidates.

The possible sentences generated for this example are *"Genetic risk **refers** more to your **chance** of inheriting a disorder or disease ."* and *"Genetic risk **refers** more to your **chants** of inheriting a disorder or disease ."*.

In this example the first sentence is the selected as the answer by the system. As can be appreciated the word *chants* just worsen the second sentence. This capacity to discriminate the wrong sentence is what draws our attention to continue with future work.

With this conditions our system produced 3613 new sentences from the original 1312. To choose the final answer from the set of proposed sentences for each sentence, we only sum all the probabilities of the syntactic tri-grams of each sentence, naturally the sentence with a higher mass of probability is the final proposed answer.

4 Evaluation

Our official results in the CoNLL 2014 Shared Task on grammatical error correction of the NUCLE and evaluated with the official scorer

(Dahlmeier and Ng, 2012) are shown in the table 8. The organizers provide all the resources.

Without alternative annotation	
Recall	2.85
Precision	11.28
F_0.5	7.09
With alternative annotation	
Recall	3.17
Precision	11.66
F_0.5	7.59

Table 8: Results in the CoNLL 2014 Shared Task .

The scoring without alternative answers was made with gold edits of the annotators and the scoring with alternative annotation includes answers proposed by 3 teams that participated on the Shared Task and were judged by the annotators.

5 Conclusions

The result of the system was not good or as we expected, first because our approach is simple and was motivated to test the use of a syntactic n-grams language model, second because the poor election of candidates to correct the errors. However, this task gave us the opportunity to test the behaviour in different conditions and now we have a reference to improve our system.

6 Future work

We have a lot of work to do, in order to support the use of this kind of resources. First we have to compare the same method that we used, but with a common n-gram language model. Second is necessary to make a more general language model that can be used with syntactic 4-grams or more, and analyse how this increase can affect the recall. Third find a way to made more efficient the consult of the resources.

Also we need to add a more wise method to correct the detected errors, including prepositions. The fact that we did not take into account this type of error does not mean that is not possible to do it with this resources, so we have to propose an alternative that takes into account this and other types of errors.

Acknowledgements

Work done under partial support of Mexican Government (CONACYT, SNI) and Instituto Politécnico Nacional, México (SIP-IPN, COFAA-IPN, PIFI-IPN).

References

Christopher Manning, Tim Grow, Teg Grenager, Jenny Finkel, and John Bauer. PTBTokenizer http://nlp.stanford.edu/software/tokenizer.shtml

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu 2013. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2013)* . (pp. 22 – 31). Atlanta, Georgia, USA.

Daniel Dahlmeier and Hwee Tou Ng 2012. Better Evaluation for Grammatical Error Correction. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2012)* . (pp. 568 – 572). Montreal, Canada.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto and Joel Tetreault. 2013. The CoNLL-2013 Shared Task on Grammatical Error Correction. *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task (CoNLL-2013 Shared Task)* .

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant 2014. The CoNLL-2014 Shared Task on Grammatical Error Correction. *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task (CoNLL-2014 Shared Task)* . Baltimore, Maryland, USA.

Leonardo Souza (leonardossz@gmail.com). 2012. Multithread-Wikipedia-Extractor for SMP based architectures, Version: 1.0 (October 15, 2012). https://bitbucket.org/leonardossz/multithreaded-wikipedia-extractor/wiki/Home

Marie Catherine de Marneffe and Christopher D. Manning. 2008. Stanford Dependencies manual.

Marie Catherine de Marneffe, Bill MacCartney and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. *In LREC 2006.*

Grigori Sidorov Book 2013. Non-linear construction of n-grams in computational linguistics: syntactic, filtered, and generalized n-grams. G. Sidorov. 2013, 166 p.

Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, Liliana Chanona-Hernández. 2014. Syntactic N-grams as Machine Learning Features for Natural Language Processing *I Expert Syst. Appl.*. vol. 41, no. 3, pp. 853-860, 2014.

Grigori Sidorov, 2013. Syntactic Dependency Based N-grams in Rule Based Automatic English as Second Language Grammar Correction. *International Journal of Computational Linguistics and Applications*. vol. 4, no. 2, pp. 169-188, 2013.

Grigori Sidorov, Anubhav Gupta, Martin Tozer, Dolors Catala, Angels Catena and Sandrine Fuentes. 2013. Rule-based System for Automatic Grammar Correction Using Syntactic N-grams for English Language Learning (L2). *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*. pp. 96-101, 2013.

Non-continuous Syntactic N-grams from Wikipedia for the CoNLL 2014 Shared Task. 2014. http://iarp.cic.ipn.mx/~dhernandez/conll2014/ http://sdavidhernandez.com/conll2014/

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing With Compositional Vector Grammars. *Proceedings of ACL 2013*

Wikipedia English dump. 2013. enwiki-20130904-pages-articles.xml.bz2 http://en.wikipedia.org/wiki/Wikipedia:Database_download