Syntactic Dependency-Based N-grams: More Evidence of Usefulness in Classification

Grigori Sidorov[1], Francisco Velasquez[1], Efstathios Stamatatos[2],
Alexander Gelbukh[1], and Liliana Chanona-Hernández[3]

[1] Center for Computing Research (CIC),
Instituto Politécnico Nacional (IPN), Mexico City,
Mexico
[2] University of the Aegean,
Greece
[3] ESIME, Instituto Politécnico Nacional (IPN), Mexico City,
Mexico
www.cic.ipn.mx/~sidorov

Abstract. The paper introduces and discusses a concept of syntactic n-grams (sn-grams) that can be applied instead of traditional n-grams in many NLP tasks. Sn-grams are constructed by following paths in syntactic trees, so sn-grams allow bringing syntactic knowledge into machine learning methods. Still, previous parsing is necessary for their construction. We applied sn-grams in the task of authorship attribution for corpora of three and seven authors with very promising results.

Keywords: Syntactic n-grams, sn-grams, syntactic paths, authorship attribution task, SVM classifier.

1 Introduction

First of all, let us clarify the term “syntactic n-grams”. Syntactic n-grams are NOT n-grams constructed by using POS tags, as one can interpret in a naive fashion. In fact, this cannot be so strictly speaking, because POS tags represent morphological information, and not syntactic data.

We propose a definition of syntactic n-grams that it is based, as its name suggests, on syntactic information, i.e., information about word relations. The main idea of our proposal is related to the method of construction of these n-grams. We suggest obtaining them by following the paths in syntactic trees, as explained in detail below. Thus, we get rid of surface language-specific information in sentences so characteristic of traditional n-grams, and maintain only persistent and pertinent linguistic information that has very clear interpretation. In our opinion, this is the way how syntactic information can be introduced into machine learning methods. Note that syntactic n-grams, though they are obtained in a different manner, keep being n-grams and can be applied practically in any task when traditional n-grams are used.

A. Gelbukh (Ed.): CICLing 2013, Part I, LNCS 7816, pp. 13–24, 2013.

Obviously, there is a price to pay for using syntactic n-grams (sn-grams). Namely, parsing should be performed for getting the mentioned syntactic paths. There are many parsers available for many languages, but parsing takes time. Also, not for all languages there are parsers at one's disposal.

An interesting question for future work is to evaluate if shallow parsing —much faster than complete parsing— is enough for obtaining syntactic n-grams of good quality. Our intuition is that for many tasks it will be sufficient.

Another interesting question for future is if syntactic n-grams allow better comparison of results between languages. Obviously, in translation some syntactic relations are changes, but many of them are maintained. So, syntactic n-grams will be more "comparable" across the languages, since they smooth the influence of language-specific surface structures.

In this paper, we apply syntactic n-grams to authorship attribution problem using three popular classifiers and compare their performance with traditional n-grams.

The rest of the paper is organized as follows: the discussion and examples of syntactic n-grams are presented in Section 2. The problem of authorship attribution is briefly introduced in Section 3. Then experimental results for authorship attribution based on syntactic n-grams are presented and compared with baseline sets of features in Section 4. Finally, conclusions are drawn.

2 Construction of SN-grams Using Syntactic Paths

Modern natural language processing very widely uses the concept of n-grams. N-grams can be composed of various types of elements: words, POS tags, characters. Usually n-grams are constructed according to the appearance of its elements in a text (in a sequential order).

Syntactic n-grams (sn-grams) are n-grams that are obtained from texts by following paths in syntactic trees. We proposed this concept and discussed some properties of syntactic n-grams in [1]. In that work, we also compared syntactic n-grams with traditional n-grams, skip-grams and Maximal Frequent Sequences that are purely statistical techniques. Note that unlike n-grams obtained with statistical techniques, syntactic n-grams have clear linguistic interpretation, while keeping the property of being n-grams, i.e., they can be applied in the same tasks as traditional n-grams.

As traditional n-grams, syntactic n-grams can be composed by various types of elements like words/stems/lemmas or POS-tags. In [1] we mentioned that syntactic n-grams of characters are impossible. Now we change our mind: syntactic n-grams of character can be constructed from syntactic n-grams of words. A question for further research is if they are useful.

In case of syntactic n-grams, another type of elements can be used for their composition: tags of syntactic relations (SR tags), like *pobj, det, xcomp*, etc. These tags are similar to POS tags in the sense that they are morphosyntactic abstraction and are obtained during previous linguistic processing.

It is important to mention that also mixed syntactic n-grams can be used that represents an interesting direction of future research. For example, in a bigram, the first component can be a word, while the second one being a POS tag or SR tag, etc. This combination, for example, can be useful in a study of subcategorization frames.

Note that sn-grams are ordered according to the syntactic path. It means that the main word always is the first element and the dependent word is the second one (and the next dependent word will be the third element in case of trigrams, etc.). So, in case of syntactic n-grams, the mixed type can be especially useful.

Resuming, sn-grams can be composed of:

- Words/stems/lemmas,
- POS-tags,
- Characters,
- SR tags (tags of syntactic relations),
- Combination of the previous ones (mixed sn-grams).

Note that as in case of traditional n-grams, auxiliary words (stop words) can be either considered or ignored in sn-grams.

Let us consider an example, a phrase taken from Jules Verne novel:

A member of the Society then inquired of the president whether Dr. Ferguson was not to be officially introduced.

Stanford parser [12] returns the following syntactic information that corresponds to the tree in Fig. 1 and Fig. 2.

```
(ROOT
  (S
    (NP
      (NP (DT A) (NN member))
      (PP (IN of)
        (NP (DT the) (NNP Society))))
    (ADVP (RB then))
    (VP (VBD inquired)
      (PP (IN of)
        (NP (DT the) (NN president)))
      (SBAR (IN whether)
        (S
          (NP (NNP Dr.) (NNP Ferguson))
          (VP (VBD was) (RB not)
            (S
              (VP (TO to)
                (VP (VB be)
                  (VP
                    (ADVP (RB officially))
                    (VBN introduced)))))))))
    (. .)))
```

The names of the syntactic relations that the parser returns as well are as following:. The format contains the relation name, the main word and its position in the sentence, the dependent word and its position in the sentence.

We do not represent graphically the relation for ROOT, but it indicates us where the syntactic tree starts.

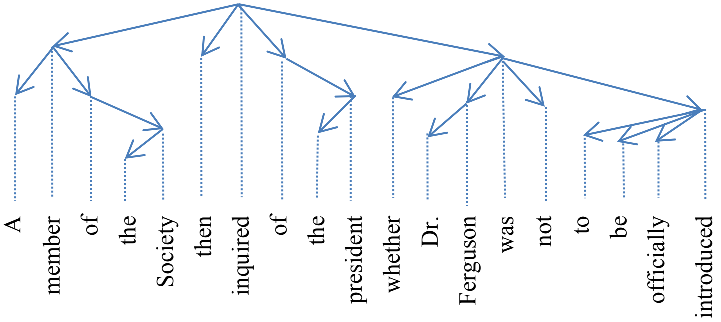

Fig. 1. Example of a syntactic tree

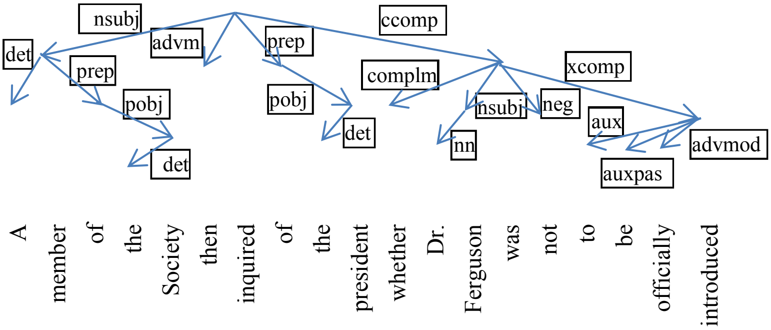

Fig. 2. Example of a syntactic tree with SR tags

det(member-2, A-1)
nsubj(inquired-7, member-2)
prep(member-2, of-3)
det(Society-5, the-4)
pobj(of-3, Society-5)
advmod(inquired-7, then-6)
root(ROOT-0, inquired-7)

prep(inquired-7, of-8)
det(president-10, the-9)
pobj(of-8, president-10)
complm(was-14, whether-11)
nn(Ferguson-13, Dr.-12)
nsubj(was-14, Ferguson-13)

ccomp(inquired-7, was-14)
neg(was-14, not-15)
aux(introduced-19, to-16)
auxpass(introduced-19, be-17)
advmod(introduced-19, officially-18)
xcomp(was-14, introduced-19)

Table 1. Character-based n-grams, baseline (3 authors)

Profile size	**Classifier**	**n-gram size**			
		2	**3**	**4**	**5**
400	SVM	**90%**	**76%**	**81%**	**81%**
	NB	71%	62%	71%	67%
	J48	76%	62%	48%	76%
1,000	SVM	**95%**	**86%**	**86%**	76%
	NB	76%	76%	67%	**81%**
	J48	81%	67%	67%	71%
4,000	SVM	**90%**	**95%**	**90%**	**86%**
	NB	**90%**	71%	81%	71%
	J48	76%	76%	**90%**	81%
7,000	SVM	NA	**90%**	**86%**	86%
	NB	NA	62%	76%	71%
	J48	NA	76%	**86%**	**90%**
11,000	SVM	NA	**100%**	**90%**	**86%**
	NB	NA	67%	62%	71%
	J48	NA	71%	81%	**86%**

Fig. 1 and Fig. 2 show graphical representation of the corresponding syntactic tree. Note that the tree reflects depth levels of each word depending on how "far" it is from the root, i.e., what is the length of the corresponding path. The arrows are drawn from main words to dependent words. In Fig. 1, we add lines that show which word corresponds to tree nodes. In Fig. 2, the tags of syntactic relations (SR tags) are represented in the squares situated as close to the corresponding arrows as possible.

We use representation of a sentence as a dependency tree. As we discussed in [1], constituency representation and dependency representation are equal for our purposes, though dependency representation in case of sn-grams is more intuitive.

We hope that these figures will help in understanding of the method of construction of sn-grams. We just traverse the tree. The method is intuitively very clear: follow the paths represented by arrows and at each step take the words (or other elements) from the corresponding nodes.

More formal description of the method is as follows. We start from the root node R, choose the first arrow (we will pass through all arrows, so the order is not important) and take the node N on the other side of the arrow. Our first bigram (or part of a larger sn-gram) is R-N. Note that the order is important because R is the main word, and N is the dependent word. Now we move to the node N and repeat the operation, either for the next bigram, or for the larger sn-gram. The number of steps

Table 2. Word based n-grams, baseline (3 authors)

Profile size	Classifier	n-gram size			
		2	3	4	5
400	SVM	**86%**	**81%**	67%	45%
	NB	48%	67%	**81%**	**85%**
	J48	67%	76%	71%	60%
1,000	SVM	**86%**	71%	71%	48%
	NB	76%	**81%**	**95%**	**90%**
	J48	71%	67%	71%	67%
4,000	SVM	**86%**	**95%**	67%	48%
	NB	62%	65%	**81%**	**86%**
	J48	81%	70%	**81%**	57%
7,000	SVM	**86%**	**90%**	71%	45%
	NB	52%	48%	81%	**81%**
	J48	**86%**	71%	**86%**	51%
11,000	SVM	**89%**	**90%**	75%	33%
	NB	53%	52%	**90%**	**78%**
	J48	**89%**	81%	70%	44%

for construction of a sn-grams of a given size is equal to *n*-1, i.e., in case of bigrams we make only one step, in case of trigrams we make two steps, etc. In bifurcations, each possible direction corresponds to a new sn-gram or a set of new sn-grams (in case that there are more bifurcations at lower levels). When we finish with a sn-gram, we return to the nearest previous bifurcation and continue in the direction that was not explored yet.

Let us compare the results for extraction of traditional bigrams and syntactic bigrams.

Traditional bigrams of words for the example are: *a member, member of, of the, the Society, Society then, then inquired, inquired of, of the, the president, president whether, whether Dr., Dr. Ferguson, Ferguson was, was not, not to, to be, be officially, officially introduced.*

Syntactic bigrams of words for the example are: *inquired member, member a, member of, of Society, Society the, inquired then, inquired of, of president, president the, inquired was, was whether, was Ferguson, Ferguson Dr., was not, was introduced, introduced to, introduced be, introduced officially.*

In our opinion, syntactic bigrams are much more stable and less arbitrary, i.e., have more chances to be repeated in other sentences. A simple example: if we add an adjective for any noun. Traditional n-grams in the near context will be changed,

Table 3. POS n-grams, baseline (3 authors)

Profile size	Classifier	n-gram size			
		2	3	4	5
400	SVM	**90%**	**90%**	**76%**	62%
	NB	67%	62%	57%	52%
	J48	76%	57%	52%	71%
1,000	SVM	**95%**	**90%**	**86%**	**67%**
	NB	76%	57%	62%	52%
	J48	71%	62%	81%	57%
4,000	SVM	NA	**100%**	**86%**	**86%**
	NB	NA	57%	62%	57%
	J48	NA	62%	67%	76%
7,000	SVM	NA	**100%**	**90%**	**86%**
	NB	NA	38%	62%	57%
	J48	NA	38%	86%	**86%**
11,000	SVM	NA	**95%**	**90%**	86%
	NB	NA	43%	48%	57%
	J48	NA	57%	86%	**90%**

but syntactic n-grams will maintain stable, only one new sn-gram will be added: Noun-Adjective.

Note that while the number of syntactic bigrams is equal to the number of traditional bigrams, the number of sn-grams when n>2 can be less than in case of traditional n-grams. It is so because traditional n-grams consider just plain combinations, while for sn-grams there should exist "long" paths. It is very clear for greater values of n. Say, for n=5, there are many n-grams in the above mentioned example, while there is only one sn-gram: *inquired member of Society the.* It is obvious that the number of n-grams and sn-grams would be equal only if the whole phrase has only one path, i.e., there are no bifurcations.

Since we mentioned that there can be sn-grams of SR tags (tags of syntactic relations) and we will use them in experiments further in the paper, we would like to present the bigrams extracted from the same example sentence*: nsubj-det, nsubj-prep, prep-pobj (2), pobj-det (2), ccomp-complm, ccomp-nsubj, ccomp-neg, ccomp-xcomp, xcomp-aux, xcomp-auxpass, xcomp-advmod.* In this case we traverse the tree as well, but instead of nodes we take the names of arrows (arcs). In this work, we consider that SR tags are comparable with POS tags: they have similar nature and the quantity of both types of elements is similar: 36 and 53 elements correspondingly.

Table 4. Sn-grams of SR tags (3 authors)

Profile size	Classifier	n-gram size			
		2	3	4	5
400	SVM	**100%**	**100%**	87%	**93%**
	NB	**100%**	93%	73%	67%
	J48	87%	67%	**93%**	73%
1,000	SVM	**100%**	**100%**	87%	**93%**
	NB	80%	67%	80%	80%
	J48	87%	67%	**93%**	73%
4,000	SVM	**100%**	**100%**	**93%**	**73%**
	NB	40%	40%	53%	60%
	J48	67%	47%	73%	73%
7,000	SVM	**100%**	**100%**	**87%**	**87%**
	NB	53%	33%	33%	73%
	J48	67%	80%	67%	73%
11,000	SVM	**100%**	**100%**	**93%**	**87%**
	NB	40%	33%	33%	60%
	J48	67%	80%	53%	73%

3 Authorship Attribution Problem

We discussed the state of the art of the authorship attribution problem in our previous work on application of syntactic n-grams for authorship attribution [1]; also see various related works on authorship attribution [7, 8, 9, 10], among many others. Here we will just briefly state the problem.

In this work, we consider it as a supervised classification problem. In this case, the authors represent possible classes. Given a set of training data (texts) of known authors, the systems should learn from them the differences between classes. It is expected that the system would learn the author style, because the thematic differences between texts are arbitrary. It is an important point to choose texts of the same genre or at least on the same theme, otherwise the system could learn the thematic classification. After this, the system should classify the test data (other texts), according to the learned model. The classifiers can use as features the mentioned types of n-grams and sn-grams.

4 Experimental Results and Discussion

In our previous work [1], we conducted experiments for proving that the concept of sn-grams is useful using as an example the task of the authorship attribution for a

Table 5. Word based n-grams, baseline, SVM (7 authors)

Profile size	Calssifier	n-gram size			
		2	3	4	5
400	SVM	76%	44%	32%	27%
1,000	SVM	68%	51%	44%	22%
4,000	SVM	73%	61%	44%	29%
7,000	SVM	78%	76%	61%	NS
11,000	SVM	78%	78%	NS	NS

Table 6. Sn-grams of SR tags, SVM (7 authors)

Profile size	Classifier	n-gram size			
		2	3	4	5
400	SVM	86%	73%	62%	51%
1,000	SVM	84%	84%	59%	51%
4,000	SVM	86%	76%	59%	62%
7,000	SVM	86%	78%	68%	62%
11,000	SVM	86%	78%	62%	62%

corpus of three authors. There we used only one classifier, SVM, and obtained superior results for sn-grams than for traditional n-grams.

In this paper, we present also the results for two more popular classifiers: Naive Bayes and J48 (tree classifier). Besides, we obtained preliminary results for sn-grams for a corpus of seven authors.

The corpus for three authors used in both studies contains texts downloaded from the Project Gutenberg web site. We used novels of three English speaking authors: Booth Tarkington, George Vaizey, and Louis Tracy, who belong more or less to the same time period. There are 13 novels of each author. We used 8 novels (60%, totally about 3.7 MB for each author) for training and 5 novels for classification (40%, totally about 2 MB for each author).

We used WEKA software for classification [11]. In our previous work, we used only one classifier, SVM. In this work we present results for three classifiers: SVM (NormalizedPolyKernel of the SMO), Naive Bayes, and J48. Several baseline feature sets are analyzed that use traditional n-grams: words, POS tags and characters.

We use the term “profile size” for representing the first most frequent n-grams/sn-grams, e.g., for profile size of 1,000 only first 1,000 most frequent n-grams are used. We tested various thresholds for profile size and selected five of them as presented in all tables that contain results.

When the table cell contains the value *NA (not available)*, it means that it was impossible to obtain the corresponding number of n-grams for our corpus. This situation is presented only with bigrams, because in general there are less bigrams than trigrams etc. It means that the total number of all bigrams is less than the given profile size. It can happen for bigrams of POS tags or characters, but not for bigrams of words.

Results for various baseline sets of features are presented in Tables 1, 2, and 3. Table 1 represents character based feature set classification. The character based features show high results for the SVM classification method, reaching 100% for a profile of 11,000 trigrams.

In Table 2, results for word based feature set are presented. It can be noticed that the best results are obtained for SVM with bigrams and trigrams of words, reaching the maximum of 95% for a profile of 4,000 trigrams.

In Table 3, the use of the features based on POS tag is presented. It can be observed that the best results (100%) are obtained using SVM for profile sizes 4,000 and 7,000.

Table 4 presents the results obtained using sn-grams of SR tags, showing substantial improvement in comparison with traditionally obtained feature sets in case of the SVM classifier. The results confirm that syntactic n-grams outperform other features in the task of authorship attribution.

In the vast majority of cases the results of the SVM classification method are better than NB and J48. We obtained the best performance with SVM always getting a 100% of accuracy with bigrams and trigrams for any profile size for sn-grams.

The only case when NB gives the same result as SVM (of 100%) is for the profile size of 400 for bigrams. Still, we consider that it is due to the fact that our topline can be achieved relatively easy because of the corpus size and the number of authors (only three). Note that in all other cases SVM got better results.

On the other hand, the tendency that sn-grams outperform traditional n-grams is preserved and allows us to get rid of the necessity to choose the threshold values, like the profile size for this corpus, for example, traditional POS trigrams got 100% for profile sizes of 4,000 and 7,000 only, while sn-grams (bigrams and trigrams) give 100% for all considered profile sizes.

We also performed preliminary experiments for the corpus of the works of seven authors built in similar way as the corpus of three authors. This task is much more difficult than for three authors because there are more classes. Our results for one baseline feature set (words) and sn-grams are presented in Tables 5 and 6. We used only SVM since we got better results before using it, but the experiments for NB and J48 should be performed as well in future. Sn-grams obtained better results in all case, only once for case of 11,000 profile of trigrams the results are the same. The interpretation of the results related to the nature and complexity of the task is a matter of future work. We plan to perform the comparison of all techniques and classifiers in near future for the corpus of seven authors.

5 Conclusions and Future Work

This paper proposed a concept of syntactic n-grams (sn-grams), i.e., n-grams that are constructed using syntactic paths. For this we just traverse the syntactic tree. The concept of sn-grams allows bringing syntactic information into machine learning methods. Sn-grams can be applied in all tasks when traditional n-grams are used.

We analyzed several properties of syntactic n-grams and presented an example of a phrase with extracted sn-grams. A shortcoming of sn-grams is that syntactic parsing is necessary prior to their construction.

We tried the concept of sn-grams in the task of authorship attribution and obtained very promising results for a corpus of three and seven authors.

In our experiments, the best results were always achieved by SVM classifier, as compared with NB and J48.

We would like to mention the following directions of future work:

- Experiments with all feature sets on larger corpus (7 authors, or more).
- Analysis of the applicability of shallow parsing instead of full parsing.
- Analysis of usefulness of sn-grams of characters.
- Analysis of behavior of sn-grams between languages, e.g., in parallel texts or comparable texts.
- Application of sn-grams in other NLP tasks.
- Application of mixed sn-grams.
- Experiments that would consider combinations of the mentioned features in one feature vector.
- Evaluation of the optimal number and size of sn-grams for various tasks.
- Consideration of various profile sizes with more granularity.
- Application of sn-grams in other languages.

Acknowledgements. Work done under partial support of Mexican government (projects CONACYT 83270, SNI) and Instituto Politécnico Nacional, Mexico (projects SIP 20111146, 20113295, 20120418, COFAA, PIFI), Mexico City government (ICYT-DF project PICCO10-120) and FP7-PEOPLE-2010-IRSES: Web Information Quality - Evaluation Initiative (WIQ-EI) European Commission project 269180.

References

1. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L.: Syntactic Dependency-Based N-grams as Classification Features. In: Mendoza, M.G. (ed.) MICAI 2012, Part II. LNCS (LNAI), vol. 7630, pp. 1–11. Springer, Heidelberg (2013)
2. Khalilov, M., Fonollosa, J.A.R.: N-gram-based Statistical Machine Translation versus Syntax Augmented Machine Translation: comparison and system combination. In: Proceedings of the 12th Conference of the European Chapter of the ACL, pp. 424–432 (2009)
3. Habash, N.: The Use of a Structural N-gram Language Model in Generation-Heavy Hybrid Machine Translation. In: Belz, A., Evans, R., Piwek, P. (eds.) INLG 2004. LNCS (LNAI), vol. 3123, pp. 61–69. Springer, Heidelberg (2004)
4. Agarwal, A., Biads, F., Mckeown, K.R.: Contextual Phrase-Level Polarity Analysis using Lexical Affect Scoring and Syntactic N-grams. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL), pp. 24–32 (2009)

5. Cheng, W., Greaves, C., Warren, M.: From n-gram to skipgram to concgram. International Journal of Corpus Linguistics 11(4), 411–433 (2006)
6. Baayen, H., Tweedie, F., Halteren, H.: Outside The Cave of Shadows: Using Syntactic Annotation to Enhance Authorship Attribution. Literary and Linguistic Computing, pp. 121–131 (1996)
7. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology 60(3), 538–556 (2009)
8. Juola, P.: Authorship Attribution. Foundations and Trends in Information Retrieval 1(3), 233–334 (2006)
9. Argamon, S., Juola, P.: Overview of the international authorship identification competition at PAN-2011. In: 5th Int. Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (2011)
10. Koppel, M., Schler, J., et al.: Authorship attribution in the wild. Language Resources and Evaluation 45(1), 83–94 (2011)
11. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
12. de Marneffe, M.C., MacCartney, B., Manning, C.D.: Generating Typed Dependency Parses from Phrase Structure Parses. In: Proc. of LREC (2006)