Commas recovery with syntactic features in French and in Czech

Christophe Cerisara* Pavel Král+ Claire Gardent*

* LORIA UMR 7503
BP 239 - 54506 Vandoeuvre
France

+Dept. of Computer Science & Engineering
University of West Bohemia
Plzeň, Czech Republic

Abstract

Automatic speech transcripts can be made more readable and useful for further processing by enriching them with punctuation marks and other meta-linguistic information. We study in this work how to improve automatic recovery of one of the most difficult punctuation marks, commas, in French and in Czech. We show that commas detection performances are largely improved in both languages by integrating into our baseline Conditional Random Field model syntactic features derived from dependency structures. We further study the relative impact of language-independent vs. specific features, and show that a combination of both of them gives the largest improvement. Robustness of these features to speech recognition errors is finally discussed.

Index Terms: Dependency parsing, punctuation detection, commas recovery

1. Introduction and related works

Automatic speech transcripts are still difficult to read, because of recognition errors, but also because of the missing structure of the document, and in particular capitalization and punctuation. We focus in this work on the task of recovering commas in a given text, which may also help subsequent automatic processing such as parsing and mining.

Punctuation recovery is often realized based on prosodic (pauses, pitch contours, energy) and lexical (surrounding words, n-grams) features, such as in [1], where full stops, commas and question marks are recovered using a finite state approach that combines lexical n-grams and prosodic features. Commas are recovered with a Slot Error Rate (SER) of 81% on automatically transcribed utterances of the Hub-4 English audio corpus. Both prosodic and lexical features are also combined via a maximum entropy model in [2], where commas are recovered on the Switchboard corpus with a F-score of 79% with lexical features only, while prosody does not help at all. For English, both the works reported in [3] and [4] (described next) show that syntactic features are very important for punctuation recovery.

Punctuation recovery has also been studied in other languages than English: In [5], automatic capitalization is realized along with automatic recovery of full stops and commas in Portuguese. Both punctuation marks are detected with a maximum entropy model that exploits acoustic and lexical features. Commas are hence recovered on automatic speech transcripts with an SER of 101%.

The authors of [6] exploit a hidden-event n-gram model combined with a prosodic model to recover punctuation marks on the Czech broadcast news corpus. F-scores of 66% and 68% are reported for commas recovery with respectively the lexical n-gram only and the lexical model combined with the decision tree model for prosody. In both previous works, no syntactic features are used.

In [7], a maximum entropy model is also exploited to recover 14 punctuation marks from the Penn Chinese TreeBank. For commas, the model exhibits a F-score of 81.14%. In order to achieve such performances, syntactic features derived from the manual syntactic annotations are used.

The authors of [4] focus on the study of comma prediction in English with syntactic features. They have compared three sequence models: Hidden-Event Language Model (HELM), factored-HELM and Conditional Random Fields (CRF). They report that the best results have been obtained with CRF, although CRFs may not scale easily to large databases.

2. Commas recovery approach

We propose to extend the work of [4] in the following aspects:

- Design of new syntactic features dedicated to comma recovery and derived from dependency structures.
- Evaluation of these features on two new languages: French and Czech.

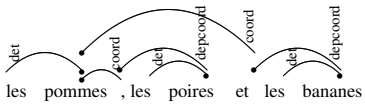
In French and Czech, the available corpora are far from being as large as in English, and scaling is not yet an issue. We have thus decided to base our work on CRF models. Furthermore, considering the relatively limited impact of prosodic features for commas recovery as reported in the literature, only lexical and syntactic features are exploited next. A CRF model is then trained to classify every subsequent word into two classes: the class of words that are followed by a comma, and the class of words without comma. The CRF input features are only local and derived from the current, previous and next words. These features are then pushed in sequence, with special words inserted at sentence boundaries, into a feature stream that is used to train the CRF model. The test corpus is processed in the same way.

3. Syntactic features

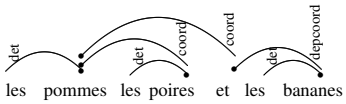
3.1. French dependency features

3.1.1. Syntactic feature for coordination

In French, commas are commonly used to serve different purposes [8]. One of their most common usage is as a replacement of coordinations, such as “et” (and) and “ou” (or). The following illustrates this usage of commas, for the nominal group “The apples, pears and bananas”:



The dependency tree is represented on top of the words, with oriented dependency arcs between the head word (circled extremity of the arc) and its governed word. Dependencies are labelled with grammatical functions, such as *det* for determiner, *coord* for coordinator and *depcoord* for coordination dependent. This example follows the annotation guidelines of the French Treebank, in which commas have an explicit role in the coordination structure. Our objective is to recover commas, which implies to remove them from the corpus first. This is achieved by automatically transforming the previous example tree into the following one, in order to preserve the coordination structure:

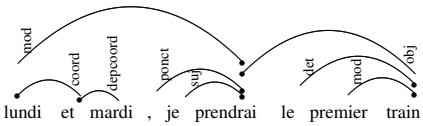


We have designed a feature to capture this usage of commas as follows: let w_i be the i^{th} word in the sentence. We want to know whether a comma shall be put between w_i and w_{i+1}. We then successively look at words w_{i+1}, w_{i+2}, $\cdots$ until we find a word w_{i+n} that is governed by a head word w_{i-m} $m \geq 0$ on the left of w_i. The feature is then TRUE iff this dependency label is "coord", otherwise, it is FALSE. This raw feature is further modified according to the identity of the following word w_{i+1}. When w_{i+1} is one of the two coordination keywords "et" (and) and "ou" (or), then the feature is set to FALSE. In the previous example, this feature is, for each word: FALSE (les), TRUE (pommes), TRUE (les), FALSE (poires), FALSE (et), FALSE (les), FALSE (bananes).

This feature is hereafter called **IsCoord**.

3.1.2. Syntactic feature for modifiers

Another common usage is to separate the modifier group that is before the verbal group, such as in "on Monday and Tuesday, I will take the first train":



We can note a few new dependencies in this example: *mod* for modifier, *ponct* for punctuation marks that are not part of a coordination structure, *suj* for subject and *obj* for object. In this example, *mardi* (Tuesday) should be followed by a comma, while *premier* (first) should not.

Intuitively, we will look at subtrees, or constituents, which are modifiers of another following word in the sentence. Then, commas may occur right after such constituents. This may be inferred, for the target word w_i, by looking for every subtree for which w_i is the rightmost word. Then, we check whether this subtree is a modifier of another word w_{i+m} $m > 0$ that is anywhere in the sentence after w_i. It is important to check that w_i is indeed the rightmost word of the modifier constituent, because commas usually only occur right after the constituent, and not within the constituent. Every time these conditions are met, the feature is defined as the "distance", i.e. the number of words between the head of this constituent and w_i.

In the previous example, "lundi" is the rightmost word of a single subtree composed of a single node (itself). The head of this subtree is thus also "lundi", which is indeed a modifier of a word at its right ("prendrai"). So the value of this feature for lundi is 0. Similarly, the feature value is 1 for "et" and 2 for "mardi". It is then -1 for "je", "prendrai" and "le", because there is no modifier, and it is 0 for "premier" and -1 for "train".

We hereafter refer to this feature as **IsMod**.

3.1.3. Syntactic feature for cross-dependencies

We propose here to generalize both previous features into a new feature that encodes cross-dependency relations between both parts of the sentence, before and after the target candidate word w_i. The intuitive idea behind this feature is that commas are more likely to separate two weakly dependent chunks than to occur within a chunk. This feature is computed as follows, for the target word w_i:

- We check whether the head of w_{i+1} is located before w_i; if so, then the corresponding dependency is crossing the limit between w_i and w_{i+1}. The value of the feature is then the label of this dependency.
- Otherwise, the same test is performed recursively for every ancestor of w_{i+1}, i.e., for the head of the head of w_{i+1}, and so on, until a crossing dependency is found or until the root of the tree is reached.
- If the root of the tree is reached without finding any crossing dependency, then we look recursively for a left-to-right crossing dependency on top of w_i, its head, etc.

In the previous example, the feature values are:
lundi(coord), et(depcoord), mardi(LEFTmod), je(LEFTsuj), prendrai(obj), le(obj), premier(obj), train(NIL)

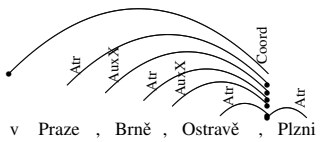
This feature is hereafter called **DepCross**. It is used next both in French and Czech experiments.

3.2. Czech dependency features

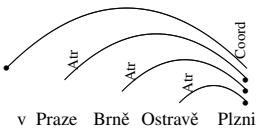
3.2.1. Syntactic feature for coordination

In the Czech language, the usage of commas is much more regular than in French, as described in [6], and commas most frequently precede specific grammatical function words, such as "protože" (because), "ale" (but), "který" (what), etc. This regularity mainly explains the relatively high F-scores reported in [6] and in our own experiments in Czech when using only contextual lexical features.

Nevertheless, another usage of commas in Czech that might be better recovered with syntactic features concerns, just like in French, coordination constructs. In the Prague Treebank, commas also play an explicit role in the syntactic tree, such as in "In Prague, Brno, Ostrava, Pilsen":



As we did for French, such structures are thus automatically transformed into:



A new syntactic feature has been specifically designed for this usage of commas in Czech. Intuitively, this feature aims at detecting coordination occurrences that involve more than three items. In such cases, just like in French, all items are usually separated by commas, except for the last two items that are separated by a coordination word, such as "a" (and) and "nebo" (or). The feature designed to detect this frequent pattern is computed as follow, for the target word w_i:

1. Recursively parse the tree branch from w_i to the root of the tree, i.e., the head of w_i, the head of its head, etc. until the root or a dependency "coord" is found (*Plzni* in the example).
2. When such a coordinator is found, list all of its direct children and finds the dependency label that occurs the most frequently amongst them (*Atr*). We assume this is the dependency type between the coordinated items and their common head.
3. The children with this dependency type are sorted chronologically and all items that occur *after* the coordination keyword are removed (none in the example).
4. For each remaining item, we check whether w_i is the right-most word of the corresponding subtree (always true in the example, as every child is composed of a single-word); if it is, then the chosen feature is the number of remaining items between this target group and the coordination keyword (e.g., 2 for *Praze*).
5. The feature is set to -1 whenever any of the preceding conditions do not hold.

This feature is hereafter called **IsCoordCz**.

3.2.2. Dependency type feature

Apart from the special case of coordination, we have further used another generic feature derived from the parse tree, which is the label of the dependency from the target word to its head. This generic feature shall cover use cases for commas that are neither handled by lexical information only, nor by a coordination structure. For example, in the previous utterance "v Praze Brně Ostravě Plzni", the features for each word are: *Root*, *Atr*, *Atr*, *Atr* and *Coord*.

This feature is hereafter called **DepLabel**. Just like **DepCross**, **DepLabel** has been tested on both French and Czech corpus, but it had no impact on the French F-score.

4. Experimental validation

4.1. Experimental set-up

The French Treebank (FTB) is composed of 12500 sentences and 325 000 words [9]. It consists of articles from *Le Monde* newspaper manually enriched with phrase structure annotations, which are further automatically converted into syntactic dependencies. The Prague Dependency Tree Bank (PDT 2.0) [10] is a collection of Czech newspaper texts that are annotated on the three following layers: morphological (2 million words), syntactic (1.5 million words) and complex syntactic and semantic layer (0.8 million words). In this work, only the syntactic dependencies of the second layer are considered. All experiments are realized in 10 folds cross-validation, where 9 tenths of the corpus is used to train the CRF model, and 1 tenth for testing.

We use a modified version of the Stanford CRF package initially developed for Named Entity Recognition [11] to train and test CRFs. The main modification concerns the possibility to complement the lexical and morphosyntactic features with the syntactic features for training and testing the CRF. The basic lexical features include (w_{i-1}, c), (w_i, c) and (w_{i+1}, c) where $c \in \{\text{comma}, \text{nocomma}\}$ is the class of word w_i. The morphosyntactic features (POS-tags) include (p_{i-1}, c), (p_i, c) and (p_{i+1}, c).

Two evaluation metrics are used: the classical F-score and the Slot Error Rate, as defined in [12].

4.2. Experimental results in French

Table 1 compares the performances of different feature sets for recovering commas on the French Treebank corpus, respectively with manual and automatic parses. The parsing is realized with our French version of the Malt Parser [13]. This parser has been trained on the very same 9 tenth of the corpus also reserved for training the comma-CRF. The CRF is actually always trained on the manual (gold) dependency trees of this 9 tenth corpus. Hence, automatic parsing is only used on the test set. Although this is a convenient approach because it does not require a double cross-validation procedure, it may be suboptimal, because the comma CRF only uses perfect dependency trees during training. On the other hand, the impact of parsing errors in table 1 is so small than it does not justify to train the CRF with parser errors.

	Manual deps.		Auto deps.	
	F-sc	SER	F-sc	SER
Lexical	38.7	93.6	38.7	93.6
Lexical + POS	43.2	85.4	**43.2**	85.4
Lexical + POS + IsCoord	46.9	82.7	46.5	83.0
Lexical + POS + IsMod	48.4	79.5	47.7	80.2
Lexical + POS + DepCross	75.0	50.7	74.9	51.1
Lexical + POS + all Synt.	76.4	47.6	**76.1**	48.4

Table 1: Comparison of different feature sets on the French Treebank. The confidence interval is $\pm 1.35\%$.

First, we can note that our baseline results are comparable to the baseline results of the state-of-the-art: hence our baseline F-score of 43.2% (for French) is comparable to the F-score of 46.9% (for English) obtained on the Gigaword corpus with lexical features in [4]. Note that there is a very large difference in corpus size between [4] and this work: 300 Kwords vs. 500 Mwords, i.e., an order of magnitude of 1000.

Second, adding any of the syntactic features helps, and their combination brings a dramatic improvement over the baseline lexical+POS features: more than +30% in F-score. Furthermore, despite parsing errors of about 15%, syntactic features still improve commas detection with automatic dependency parses by more than 30% absolute.

4.3. Experimental results in Czech

Table 2 compares different feature sets on the Prague Treebank, respectively with manual and automatic parsing.

	Manual deps.		Auto deps.	
	F-sc	SER	F-sc	SER
Lexical	62.9	62.0	62.9	62.0
Lex. + POS	64.1	56.0	**64.1**	56.0
Lex. + POS + IsCoordCz	69.6	49.6	66.6	53.1
Lex. + POS + DepLabel	77.1	39.6	77.0	40.5
Lex. + POS + DepCross	79.6	36.3	78.2	38.1
Lex. + POS + AllSynt	91.2	16.8	**85.5**	27.0

Table 2: Comparison of different feature sets on the Prague Treebank. The confidence interval is $\pm 0.1\%$.

In Czech, the baseline performances are much higher than in French, as already discussed in section 3.2. The syntactic feature dedicated to handle coordination brings largely significant improvements, about +2.5% absolute. The best feature in Czech is DepCross (+14%), as in French experiments.

Despite parsing errors of about 34%, syntactic features still improve commas detection F-score by +21.4% in absolute value. This clearly confirms the effective importance of syntactic features to recover commas.

5. Discussion

5.1. Feature dependency to the language

Our initial objective in this work was to design generic syntactic features that could be applied to different languages, similarly to the basic word form and part-of-speech tag features, which are applied as is in most natural language processing tasks. This objective has only been partially reached. Indeed, we have observed that the most "basic" syntactic features, such as **DepLabel**, may be very effective in some languages but not on others. We nevertheless proposed such a generic feature, **DepCross**, which seems to work very well in both languages.

Aside from this quest for generic features, we also focused our efforts towards addressing specific usages of commas, such as coordination or modifiers, where syntactic information might intuitively bring valuable information. This approach also gave some improvement in both languages, at the cost of devising and implementing much more complex syntactic features. Nevertheless, such focused features might prove useful, as shown in our experiments, because they address specific patterns that may not be correctly handled by generic features only.

5.2. Robustness to speech recognition errors

Punctuation recovery is a typical language processing task that can be applied to automatic speech transcriptions. One might question the robustness of the proposed syntactic features to speech recognition errors, because of the known limited performances of syntactic parsers on automatic transcriptions. We have not been able so far to test our system on such speech transcriptions, because the French and Prague treebanks are both written text corpora. Furthermore, testing our system on another speech corpus, such as the ESTER corpus, would first require to develop an efficient parser on this type of data. Indeed, domain adaptation of syntactic parsers is known to be extremely difficult, as demonstrated in the CoNLL'2007 campaign, which prevents a direct application of written-text parsers to such corpora. Although we have recently made some progress in this direction [13], there is still no satisfying existing parsing solution nor resources for French spoken data.

6. Conclusions and future work

This work extended previous works dedicated to commas recovery, and in particular [4]. Two new languages are considered, and syntactic features are derived from the dependency tree for each of them. In both cases, the syntactic features improve the performances largely above significance levels. This supports the published conclusions on the importance of syntax for this task and extends them to French and Czech. The next steps will consist in extending this work to support automatic speech recognition outputs, with the objective of enriching such transcripts with punctuation. However, this requires first to solve the weakness of nowadays French and Czech parsers, which are not robust enough to recognition errors.

7. Acknowledgements

This work has been partly supported by the Région Lorraine and the CPER MISN TALC, by a project partly funded by the European Commission. We also thank Mr. Michal Hrala for his help for some experiments.

8. References

[1] H. Christensen, Y. Gotoh, and S. Renals, "Punctuation annotation using statistical prosody models," in *Proc. ISCA Workshop on Prosody in Speech Recognition and Understanding*, 2001, pp. 35–40.

[2] J. Huang and G. Zweig, "Maximum entropy model for punctuation annotation from speech," in *Proc. ICSLP*, 2002, pp. 917–920.

[3] S. M. Shieber and X. Tao, "Comma restoration using constituency information," in *Proc. HLT-NAACL*, 2003, pp. 142–148.

[4] B. Favre, D. Hakkani-Tür, and E. Shriberg, "Syntactically-informed models for comma prediction," in *Proc. ICASSP*, Taipei, Taiwan, April 2009, pp. 4697–4700.

[5] F. Batista, D. Caseiro, N. Mamede, and I. Trancoso, "Recovering capitalization and punctuation marks for automatic speech recognition: Case study for Portuguese broadcast news," *Speech Communication*, vol. 50, pp. 847–862, 2008.

[6] J. Kolář, J. Švec, and J. Psutka, "Automatic punctuation annotation in Czech broadcast news speech," in *Proc. SPECOM*. Saint-Petersburg: SPIIRAS, 2004, pp. 319–325.

[7] Y. Guo, H. Wang, and J. v. Genabith, "A linguistically inspired statistical model for Chinese punctuation generation," *ACM Transactions on Asian Language Information Processing*, vol. 9, no. 2, p. 27, 2010.

[8] M. Simard, "étude de la distribution de la virgule dans les phrases de textes argumentatifs d'expression française," Ph.D. dissertation, Univ. du Québec Chicoutimi, Apr. 1993.

[9] M.-H. Candito, B. Crabbé, and P. Denis, "Statistical French dependency parsing: treebank conversion and first results," in *Proc. LREC*, La Valletta, Malta, 2010.

[10] J. Hajič, A. Böhmová, E. Hajičová, and B. Vidová-Hladká, "The Prague dependency treebank: A three-level annotation scenario," in *Treebanks: Building and Using Parsed Corpora*, A. Abeillé, Ed. Amsterdam:Kluwer, 2000, pp. 103–127.

[11] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," in *Proc. Association of Computational Linguistics*, 2005, pp. 363–370.

[12] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, "Performance measures for information extraction," in *Proc. DARPA Broadcast News Workshop*, Herndon, VA, 1999.

[13] C. Cerisara and C. Gardent, "Analyse syntaxique du français parlé," in *Journée thématique ATALA : Quels analyseurs syntaxiques pour le français ?*, 2009.