Correcting prepositional phrase attachments using multimodal corpora

Sebastien Delecraz and **Alexis Nasr** and **Frederic Bechet** and **Benoit Favre**
Aix-Marseille University, CNRS, LIF
firstname.lastname@lif.univ-mrs.fr

Abstract

PP-attachments are an important source of errors in parsing natural language. We propose in this article to use data coming from a multimodal corpus, combining textual, visual and conceptual information, as well as a correction strategy, to propose alternative attachments in the output of a parser.

1 Introduction

Prepositional phrase attachments (*PP-attachments*) are known to be an important source of errors in parsing natural language. The main reason being that, in many cases, correct attachments cannot be predicted accurately based on pure syntactic considerations: their prediction ask for precise lexical co-occurrences or non linguistic knowledge. Such information is usually not found in treebanks that are limited in their size and therefore do not model many bi-lexical phenomena.

In this paper, we propose to combine textual, conceptual and visual information extracted from a multimodal corpus to train a PP-attachment correction model. In order to do so, we have used a corpus made of pairs (S, P) where S is a sentence and P a picture. Some words of S have been manually linked to bounding boxes in P and tagged with coarse-grained conceptual types. The relative positions of the boxes in the pictures as well as conceptual types and the lexical nature of words involved in a PP-attachment are used as features for a classifier that classifies a PP-attachment as either correct or wrong. Given the parse tree T of S, and a target preposition, its different possible attachment sites are identified and the classifier is used to select the most promising one.

Our contributions in this study are the selection and the manual annotation of a corpus of ambiguous PP-attachments from the multimodal corpus *Flickr30k Entities* (Plummer et al., 2017); the study of the relative importance of different kinds of features for the PP-attachment resolution problem, from very specific ones (lexical features) to very generic ones (spatial features); and the combination of them in a single model for improving the accuracy of a syntactic dependency parser.

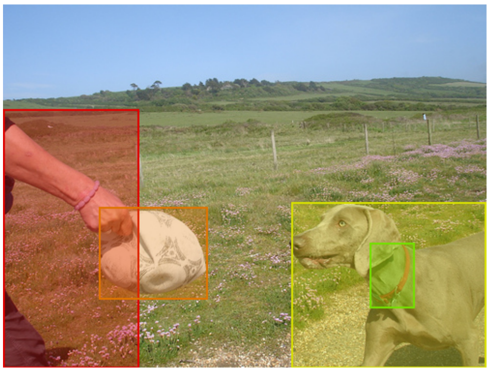
The structure of the paper is the following: section 2 presents some related work in the fields of PP-attachment and multimodal language processing. In section 3 the multimodal corpus is described as well as the manual annotation that has been performed on it. In section 4 the error prediction classifier is described and its performance evaluated. In section 5, the correction strategy is described. The experiments are described in section 6 and section 7 concludes the paper.

2 Related work

This work is related to two different areas: *multimodal language processing* through the joint analysis of image and text describing the same scene and *syntactic parsing* through the problem of prepositional phrase attachment resolution (*PP-attachments*).

The joint processing of image and natural language is not novel. It has been studied mostly in the context of natural language generation, for example for generating a textual description of a video or an image. Early work (Herzog and Wazinski, 1994) computes first spatial relations among objects detected in images with knowledge-based language generation model in order to generate short descriptions of videos in limited domains (traffic scenes, soccer matches). Recently open-domain language generation from

Proceedings of the 15th International Conference on Parsing Technologies, pages 72–77,
Pisa, Italy; September 20–22, 2017. ©2017 Association for Computational Linguistics



1. someone is holding out a punctured ball in front of a brown dog with a red collar .
2. A man holding out a deflated soccer ball to a gray dog .
3. The owner tries to hand a deflated ball to his dog .
4. Large gray dog being handed a white soccer ball .
5. A brown dog starring at a soccer ball .

Figure 1: Example of the *F30kE* annotations.

images or videos received a lot of attention through the use of multimodal deep neural networks (Vinyals et al., 2015). Theses models built a unified representation for both image and language features and generate in an end-to-end process a text directly from an image, without an explicit representation (syntactic or semantic) of the text generated.

For syntactic parsing, the problem of PP-attachment has a long history in Natural Language Processing and a wealth of different methods and sources of information have been used to alleviate it. Giving a overview of this vast body of literature is well beyond the scope of this paper. Traditionally, two types of resources have been used to help resolving PP-attachment, semantic knowledge bases (Agirre et al., 2008; Dasigi et al., 2017), and corpora (Rakshit et al., 2016; Mirroshandel and Nasr, 2016; Belinkov et al., 2014; de Kok et al., 2017).

We are not aware of much work using multimodal information for PP-attachment. In the most relevant work that we have found (Christie et al., 2016), a parser is used to predict the k best parses for a sentence and this set is re-ranked using visual information. The main difference with their work is, in our case, the combined use of lexical, semantic and visual cues as well as the method used (k best parses v/s parse correction).

3 Data

The multimodal corpus used in this work is the Flickr30k Entities (*F30kE*) (Plummer et al., 2017), an extension of the original Flickr30k dataset (Young et al., 2014). This corpus is composed of almost $32K$ images and, for each image, five captions describing the image have been produced. Besides, every object in the image that corresponds to a mention in the captions has been manually identified with a bounding box. Bounding boxes and the mentions in the captions have been paired together via co-reference links. A total of $244K$ such links have been annotated. Furthermore, each mention in the captions has been categorized into eight coarse-grained conceptual types using manually constructed dictionaries. The types are: people, body parts, animals, clothing, instruments, vehicles, scene, and other. One example of the corpus has been reproduced in Figure 1.

Our goal in this study is to evaluate several set of features, at the lexical, conceptual and vision levels, for the PP-attachment task. The *F30kE* corpus contains already all these features, but no syntactic annotation was provided on the image captions. Focusing on the PP-attachment problem, we added such annotations with the following process: first the whole caption corpus of *F30kE* was processed by a Part-Of-Speech tagger (Nasr et al., 2011); a set of regular expressions on the POS labels were defined in order to select sentences that contain a preposition that might lead to an ambiguous PP-attachment; finally all these sentences were manually processed in order to attach the selected prepositions to their correct syntactic governor.

Captions containing ambiguous PP-attachment have been identified using two simple rules: a preposition is considered ambiguous if it is preceded by at least two nouns or a verb and a noun, in other word, the captions must match one of the following regular expressions: `X* N X* N X* p X*` or `X* V X* N X* p X*`, where `N` and `V` stand for the POS tags noun and verb, `X` stand for any POS tag and `p` is the target preposition.

22800 captions were selected this way. They constitute our *PP-corpus*. This corpus contains 29068 preposition occurrences that have been manually attached to their syntactic governor. The *PP-corpus* has been divided into a train set, made of 18241 captions (23254 annotated prepositions), a development set, made of 2271 captions (2907 annotated prepositions) and a test set, made of 2288 captions (2907 prepositions).

4 Error Prediction

The train part of the *PP-corpus* has been used to train a classifier that predicts whether a PP-attachment proposed by a parser is correct or not. The parser used is a standard arc-eager transition based parser (Nivre, 2003), trained on sections $0-18$ of the Penn Treebank (Marcus et al., 1993). The parser was run on the train set of the corpus and, for each occurrence of a manually attached preposition, a negative or a positive example has been produced depending on whether the parser has predicted the correct attachment or not. This data set is composed of 17643 positive and 5611 negative examples. It has been used to train a classifier that predicts whether the attachment made by the parser is correct or not.

The classifier used for this task is the *Icsiboost* classifier (Favre et al., 2007). This Adaboost classifier is a combination of weak learners that learn a threshold for continuous features, and a binary indicator for discrete ones. Training minimizes the exponential loss function by greedily selecting the best classifier and re-weighting the training set to focus on misclassified examples. This kind of classifier has two benefits: models are easier to interpret than in other families of models, and the greedy selection of classifier effectively selects relevant features and is less affected by noise.

Three sets of features have been used to train the classifier, from the most specific ones (lexical features) to the most generic ones (spatial features). The set T is composed of textual features, extracted from the captions. The set C is composed of conceptual features, based on the conceptual classes associated with boxes or words. Set V is composed of visual features representing spatial information about the objects annotated in the image (enriched with bounding boxes).

Let GpD be a PP-attachment where G is the governor of the preposition p and D its dependent. We define the functions $POS(X)$ that denote the POS of word X, $LEM(X)$ its lemma, $FCT(X)$ its syntactic function, $CON(X)$ the list of its conceptual types, $BB(X)$ its bounding box (in the case where X is associated with several boxes, $SBB(X)$ lists this set). $DIST(X,Y)$ represents the distance between words X and Y in the sentence. Here is a detailed description of the features of the three different categories using the notations defined above[1]:

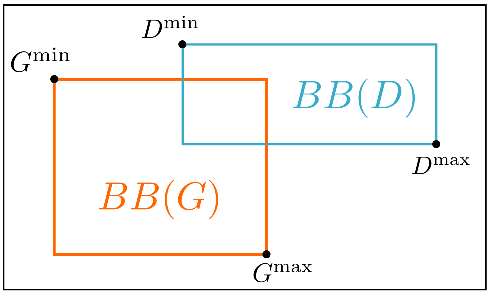


Figure 2: Points of interest from (G) and (D) boxes.

Textual features:

$T_1 = POS(G)$ $\quad T_2 = LEM(G)$
$T_3 = POS(D)$ $\quad T_4 = LEM(D)$
$T_5 = FCT(p)$ $\quad T_6 = DIST(G,p)$
$T_7 = LEM(G) + LEM(D)$
$T_8 = POS(G) + POS(D)$

Conceptual features:

$C_1 = CON(G)$ $\quad C_2 = CON(D)$
$C_3 = CON(G) + CON(D)$ Eight types of concept are defined: people, body parts, animals, clothing, instruments, vehicles, scene, and other. The value UNK is used if either G or D is not associated with a type.

Visual features: in Figure 2, we identify two corners for every bounding box B: $B^{\min}$ and $B^{\max}$, that are used compute the visual features:

$$V_1 = \frac{D_x^{\min} - G_x^{\min}}{G_x^{\max} - G_x^{\min}} \qquad V_2 = \frac{D_x^{\max} - G_x^{\min}}{G_x^{\max} - G_x^{\min}}$$
$$V_3 = \frac{D_y^{\min} - G_y^{\min}}{G_y^{\max} - G_y^{\min}} \qquad V_4 = \frac{D_y^{\max} - G_y^{\min}}{G_y^{\max} - G_y^{\min}}$$
$$V_5 = |SBB(D)| \qquad V_6 = |SBB(G)|$$
$$V_7 = Area(BB(D)) \ / \ Area(BB(G))$$

Features $V_1 \dots V_4$ describe the relative position of D box with respect to G box, respectively on the x and y axis. Features V_5 and V_6 describe the number of boxes associated with D and G. V_7 is the ratio of the areas of D and G boxes. In case of multi-boxing, we compute the distance between $G^{\min}$ and $D^{\min}$ and keep the two closest boxes as $BB(G)$ and $BB(D)$. When either G or D does not have a box, the UNK value is used.

It is important to notice that the visual features in our study are limited to spatial information about bounding boxes. No image analysis of the content of the boxes is done since this level of

[1] All feature sets also contain the general feature $LEM(p)$: the lemma of the preposition.

Features	Train	Dev	Test
Baseline	0.76	0.76	0.75
T	0.94	0.90	0.88
C	0.83	0.84	0.83
V	0.78	0.79	0.77
T + C	0.96	0.91	0.90
T + C + V	0.98	0.91	0.89

Table 1: Classifier accuracy by model.

information is covered by the *conceptual features* which attach to each box a concept tag related to its content.

Table 1 details the classification accuracy for each model trained using different feature combinations, on train, development and test sets. The accuracy is computed on the attachments predicted by the parser. A baseline has been added to Table 1 that selects the majority class which is the *positive* class, therefore it reflects the accuracy of the parser for PP-attachment (75% accuracy on the test set).

As one can see, all four models beat the baseline. The best features are the lexical ones. This is expected as they are the most specific ones, requiring a training corpus matching closely the application domain. Conceptual features obtain very good results although they can be considered as generic since only 8 types of concepts are considered. The visual features are just slighlty better than the baseline (+2%), however we have to keep in mind that the only information considered here are spatial features of bounding boxes. Since not all prepositions in the *PP-corpus* are related to spatial positions, and considering the genericity of the features used, obtaining an accuracy of 77% without any lexical or semantic features is an interesting result.

By combining feature sets we can improve accuracy. The best combination is the textual and the conceptual features together.

5 Correction Strategy

The classifier developed in the previous section only checked if a PP-attachment proposed by the parser is correct or not. In this section we integrate this classifier in a correction strategy in order to improve the accuracy of our parser. This correction strategy is inspired from the ideas of Anguiano and Candito (2011); Attardi and Ciaramita (2007); Hall and Novák (2005): given a sentence S, a parse T for S and a target preposition p, a set G_p of candidate governors for p is identified. The highest scoring $c \in G_p$ is then assigned as the new governor of p in T.

The set G_p is initialized with g, the actual governor of p in the parse T. The following rules are then applied to T and new potential governors are added to G_p:

1 $N \leftarrow V \rightarrow p \quad \Rightarrow G_p = G_p \cup \{N\}$
2 $N \leftarrow P \leftarrow V \rightarrow p \quad \Rightarrow G_p = G_p \cup \{N\}$
3 $N' \leftarrow N \rightarrow p \quad \Rightarrow G_p = G_p \cup \{N'\}$
4 $N' \leftarrow P \leftarrow N \rightarrow p \quad \Rightarrow G_p = G_p \cup \{N'\}$
5 $N' \rightarrow X \rightarrow N \rightarrow p \quad \Rightarrow G_p = G_p \cup \{N'\}$
6 $N \rightarrow N \rightarrow p \quad \Rightarrow G_p = G_p \cup \{N\}$
7 $V \rightarrow N \rightarrow p \quad \Rightarrow G_p = G_p \cup \{V\}$

Rule 1 is interpreted as follows: if target preposition p has a verbal governor which has a noun N as a direct dependent, N is added as a candidate governor. These rules have been evaluated on our development corpus. When applying the rules to the output of the parser, the correct governor of the manually annotated prepositions is in the set G_p in 92.28% of the cases. This figure is our upper bound for PP-attachments.

Given the sentence *a man throws a child into the air at a beach*, and target preposition *at* that the parser has attached to *child*, the two rules 4 and 7 apply, yielding $G_p = \{child,\ air,\ throws\}$

	man	throws	child	into	air	at
4			N	P	$N*$	p
7		$V*$	N			p

The correction strategy is the following: given an attachment GpD produced by the parser, this attachment is given as input to the error detector. If the detector predicts the CORRECT class, then the attachment is kept unchanged. Otherwise, the set G_p is computed and the element g of the set that maximizes the score $S(gpD, \texttt{CORRECT})$ is selected (*i.e.* the score that the classifier associates with the class CORRECT to the given input).

6 Experiments

The results of our experiments on the test set are detailed in Table 2. The table shows the attachment accuracy for the prepositions that appear at least 30 times in the corpus. For each of these prepositions, column two displays its number of occurrences, column three (BL) shows the attachment accuracy for this preposition in the output the parser. Columns four (T), five (C), six (V) and seven (TCV) show the attachment accuracy for the corrected output for four different configurations

Prep	Occ	BL	T	C	V	TCV
into	116	0.89	0.93	0.88	0.89	0.96
with	310	0.65	0.78	0.75	0.66	0.79
through	145	0.95	0.96	0.95	0.95	0.97
behind	35	0.74	0.86	0.83	0.77	0.89
under	58	0.84	0.84	0.86	0.84	0.86
down	41	0.63	0.73	0.63	0.44	0.68
in	369	0.76	0.84	0.81	0.76	0.85
in front of	51	0.90	0.88	0.90	0.90	0.90
outside	35	0.63	0.74	0.74	0.69	0.74
on	143	0.85	0.90	0.89	0.85	0.91
around	59	0.73	0.81	0.73	0.71	0.83
for	168	0.73	0.82	0.77	0.72	0.80
at	63	0.84	0.86	0.90	0.84	0.90
along	50	0.52	0.86	0.76	0.52	0.88
across	49	0.88	0.96	0.88	0.88	0.96
against	31	0.77	0.94	0.84	0.77	0.94
near	159	0.33	0.84	0.83	0.58	0.84
towards	30	0.90	0.93	0.87	0.90	0.90
next to	137	0.89	0.89	0.88	0.89	0.89
by	76	0.84	0.86	0.86	0.84	0.87
of	72	0.93	0.93	0.93	0.93	0.93
over	111	0.66	0.85	0.73	0.66	0.86
during	41	0.71	0.76	0.73	0.71	0.76
from	140	0.76	0.86	0.78	0.76	0.84
TOTAL	2907	0.75	0.85	0.82	0.77	0.86

Table 2: PP-attachment accuracy on the test set per preposition (only those with at least 30 occurrences).

Features	Accuracy
Baseline	0.75
T	0.85
C	0.82
V	0.77
T + C	0.86
T + V	0.86
C + V	0.82
T + C + V	0.86

Table 3: PP-attachment accuracy on the test set.

of the error detector: using only one type of features (Textual, Conceptual and Visual) and all features. The last line gives the attachment accuracy on all preposition.

As one can see on Table 2, the global accuracy of the parser on all PP-attachment is equal to 75%. This figure is lower than the 86% correct PP-attachment reported by Anguiano and Candito (2011) on the Penn Treebank using the same kind of parser, which does not come as a surprise, given the different nature of these two corpora. Table 3 presents the accuracy of PP-attachment after correction with different feature set combinations. Adding conceptual features to textual features improve accuracy, however spatial features have no impact when used in conjunction with other feature sets.

Several conclusions can be drawn from these results: different prepositions have very different accuracy with the parser, ranging from 95% for preposition *through*, to 33% for preposition *near*. The correction strategy implemented has a positive impact on accuracy: changing some attachments proposed by the parser using an error corrector based on limited but specific data is useful. Similarly as the results obtained on the classification accuracy, textual features are the most useful ones. Used alone, they increase accuracy by 10 points. Although improving accuracy by 2% when used alone, visual features have no impact when combined with other feature sets. The positive impact of visual features is concentrated on three prepositions in table 2: (*near*, *behind* and *outside*). It is interesting to note that these prepositions are mostly locative. It does therefore make sense that visual features only focusing on spatial information have some impact on these prepositions. On the other extreme, preposition like *during* that are mostly temporal are logically not impacted by the correction.

7 Conclusion

We have proposed in this paper an error correction strategy for PP-attachment that extracts from a multimodal corpus features that help predict such attachments as either correct or not. This classifier is used to select among the different possible attachment points of a preposition the highest scoring one with respect to the classifier. Experiments showed that this method increases by 11 absolute points the correct PP-attachment rate. As expected the most relevant feature set is the lexical one, which is the most specific one. Conceptual features, although quite generic, obtain results close to lexical features. Visual features, limited in our case to spatial information, can improve greatly the accuracy of pp-attachment when used alone for some locative preposition, however they have no impact when mixed with more specific features.

We intend to extended this work in many directions. The first one is the definition of better visual features. We believe that more useful information can be extracted from the image to improve PP-attachment. We also consider defining a better correction strategy that will identify more possible governors to prepositions and, finally, introduce the new features directly in the parsing model.

Acknowledgments

This work has been carried out thanks to the support of French DGA in partnership with Aix-Marseille University as part of the "Club des partenaires Défense".

References

Eneko Agirre, Timothy Baldwin, and David Martinez. 2008. Improving parsing and pp attachment performance with sense information. In *ACL*. pages 317–325.

Enrique Henestroza Anguiano and Marie Candito. 2011. Parse correction with specialized models for difficult attachment types. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1222–1233.

Giuseppe Attardi and Massimiliano Ciaramita. 2007. Tree revision learning for dependency parsing. In *HLT-NAACL*. pages 388–395.

Yonatan Belinkov, Tao Lei, Regina Barzilay, and Amir Globerson. 2014. Exploring compositional architectures and word vector representations for prepositional phrase attachment. *Transactions of the Association for Computational Linguistics* 2:561–572.

Gordon Christie, Ankit Laddha, Aishwarya Agrawal, Stanislaw Antol, Yash Goyal, Kevin Kochersberger, and Dhruv Batra. 2016. Resolving language and vision ambiguities together: Joint segmentation & prepositional attachment resolution in captioned scenes. *arXiv preprint arXiv:1604.02125* .

Pradeep Dasigi, Waleed Ammar, Chris Dyer, and Eduard Hovy. 2017. Ontology-aware token embeddings for prepositional phrase attachment. *arXiv preprint arXiv:1705.02925* .

Daniël de Kok, Jianqiang Ma, Corina Dima, and Erhard Hinrichs. 2017. Pp attachment: Where do we stand? *EACL 2017* page 311.

Benoit Favre, Dilek Hakkani-Tür, and Sebastien Cuendet. 2007. Icsiboost. http://code.google.come/p/icsiboost.

Keith Hall and Václav Novák. 2005. Corrective modeling for non-projective dependency parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*. Association for Computational Linguistics, pages 42–52.

Gerd Herzog and Peter Wazinski. 1994. Visual translator: Linking perceptions and natural language descriptions. *Artificial Intelligence Review* 8(2):175–187.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics* 19(2):313–330.

Seyed Abolghasem Mirroshandel and Alexis Nasr. 2016. Integrating selectional constraints and subcategorization frames in a dependency parser. *Computational Linguistics* .

A. Nasr, F. Béchet, J.F. Rey, B. Favre, and J. Le Roux. 2011. Macaon: An nlp tool suite for processing word lattices. *Proceedings of the ACL 2011 System Demonstration* pages 86–91.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT*. Citeseer.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2017. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *International Journal of Computer Vision* 123(1):74–93.

Geetanjali Rakshit, Sagar Sontakke, Pushpak Bhattacharyya, and Gholamreza Haffari. 2016. Prepositional attachment disambiguation using bilingual parsing and alignments. *arXiv preprint arXiv:1603.08594* .

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 3156–3164.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics* 2:67–78.