The Alpage Architecture at the SANCL 2012 Shared Task: Robust Pre-Processing and Lexical Bridging for User-Generated Content Parsing

Djamé Seddah, Benoît Sagot, Marie Candito

Robust Pre-Processing and Semi-Supervized Lexical Bridging for User-Generated Content Parsing

Djamé Seddah[1,2] **Benoît Sagot**[2] **Marie Candito**[2]
1. ISHA, Université Paris–Sorbonne, Paris, France
2. Alpage, INRIA & Université Paris–Diderot, Paris, France
djame.seddah@paris-sorbonne.fr, benoit.sagot@inria.fr, marie.candito@linguist.jussieu.fr

Abstract

We describe the architecture we set up during the SANCL shared task for parsing user-generated texts, that deviate in various ways from linguistic conventions used in available training treebanks. This architecture focuses in coping with such a divergence. It relies on the PCFG-LA framework (Petrov and Klein, 2007), as implemented by Attia et al. (2010). We explore several techniques to augment robustness: (i) a lexical bridge technique (Candito et al., 2011) that uses unsupervised word clustering (Koo et al., 2008); (ii) a special instanciation of self-training aimed at coping with POS tags unknown to the training set; (iii) the wrapping of a POS tagger with rule-based processing for dealing with recurrent non-standard tokens; and (iv) the guiding of out-of-domain parsing with predicted part-of-speech tags for unknown words and unknown (word, tag) pairs. Our systems ranked second and third out of eight in the constituency parsing track of the SANCL competition.

1 Introduction

Complaining about the lack of robustness of statistical parsers whenever they are applied on out-of-domain text has almost became an overused *cliché* over the last few years. It remains true that such parsers only perform well on texts that are comparable to their training corpus, especially in terms of genre. As noted by Foster (2010; Foster et al. (2011), most studies on out-of-domain statistical parsing have been focusing mainly on slightly different newswire texts (Gildea, 2001; McClosky et al., 2006b; McClosky et al., 2006a), biomedical data (Lease and Charniak, 2005; McClosky and Charniak, 2008) or balanced corpora mixing different genres (Foster et al., 2007). The common point between these corpora is that they are edited texts, meaning that their underlying syntax, spelling, tokenization and typography remain standard, even if they diverge slightly from the newswire genre.[1] Therefore, standard NLP tools can be used on such corpora.

Now, new forms of electronic communication have emerged in the last few years, namely social media and Web 2.0 communication media, either synchronous (micro-blogging) or asynchronous (forums), thus the need for comprehensive ways of coping with the new languages types carried by those media is becoming of crucial importance.

If those unlimited stream of texts were all written with the same level of proficiency than our canonical treebanks, the problem, as it should be called, would be *simply*[2] a matter of domain adaptation. However, as shown by the challenges experienced during the *Parsing the Web* shared task (Petrov and McDonald, 2012), this is far from being the case. In this paper, we describe the architecture we set up in order to cope with prevalent issues in user generated content, such as lexical data sparseness and noisy input. This architecture relies on the PCFG-LA framework (Petrov and Klein, 2007), as implemented by Attia et al. (2010). We explore several techniques to augment robustness: (i) a lexical bridge technique (Candito et al., 2011) that uses unsupervised word clustering (Koo et al., 2008); (ii) a special instanciation of self-training aimed at coping with POS tags unknown to the training set; (iii) the wrapping of a POS tagger with rule-based processing

[1] Putting aside speech specificities, present in some known data sets such as the BNC (Leech, 1992).

[2] Cf. (McClosky et al., 2010) for numerous evidences of the non-triviality of that task.

for dealing with recurrent non-standard tokens; and (iv) the guiding of out-of-domain parsing with predicted part-of-speech tags for unknown words and unknown (word, tag) pairs.

2 System Description

The architecture we built is roughly based on the parallel processing of two streams of data: After a preprocessing stage, one stream is clusterized using the Brown algorithm (Brown et al., 1992) as implemented by Liang (2005), the other is tagged using the MELT tagger (Denis and Sagot, 2009). Only POS tags assigned to unseen (word, tag) pairs are kept. Both streams are then merged and parsed with a product of self-trained PCFG-LA grammars produced by the LORG parser (Attia et al., 2010).

2.1 POS tagging for noisy data

Given the expected noisiness of web texts, we used a specifically developped pre-annotation process. This is because in such corpora, spelling errors are extremely frequent, but also because the original tokens rarely match sound linguistic units. The idea underlying this pre-processing is to wrap a POS tagger in such a way that it actually has to tag a sequence of tokens that is as close as possible to standard English, or, rather, to its training corpus. Hence the following process:

1. An Ontonote/PTB token normalization stage is applied. Neutral quotes are disambiguated, following (Wagner et al., 2007).

2. We then apply several regular-expression-based grammars taken from the SxPipe preprocessing chain (Sagot and Boullier, 2008) for detecting smileys, URLs, e-mail addresses and similar entities, in order to consider them as one token even if they contain whitespaces. SxPipe is able to keep track of the original tokenization, which is required for restoring it at the end of the process.

3. Then, we correct tokens or token sequences according to one of the two following techniques: (i) fuzzy matching with the EnLex lexicon (a freely available English lexicon containing 500,000 entries, (Sagot, 2010)) using only very simple and frequent systematic spelling error patterns; (ii) lowercasing for uppercase-only sentences, except for the first character; (iii) a few dozens of manually crafted rewriting rules for dealing with frequent amalgams such as *gonna* or *im*.[3] This step and the previous step (steps 2 and 3) might modify the number of tokens. In such cases, we use n-to-m mappings between original and "corrected" tokens. For example, the rule *alot* $\rightarrow$ *a␣lot* explicitely states that *alot* is an amalgam for *a* and *lot*.

4. We trained a perceptron-based version of the MElt tagger (Denis and Sagot, 2009) on the Ontonotes training set, using EnLex as external lexicon; we apply the resulting tagger on the sequence of corrected tokens.

5. We assign POS tags to the orignal tokens based on the mappings between original tokens and POS-tagged corrected tokens (corrected either at step 2 or 3). If the mapping is not 1-to-1, specific heuristics are used that involve the `AFX` and `GW` tags. Additional heuristics are used for post-correcting several tag assignments (e.g., long sequences of punctuation characters are re-tagged `NFP`; URLs and e-mail addresses are tagged `ADD`, and so on).

We have conducted preliminary evaluation experiments on the MElt POS-tagger when embedded within this normalization and correction wrapper. In Table 1, we provide POS-tagging accuracy results over the three development sets provided before the shared task: the Ontonotes development set as well as the e-mail and weblog development corpora. The results indicate that using the normalization and correction wrapper leads to significant improvements in POS tagging accuracy for the e-mail corpus, without harming performances on higher-quality corpora (it actually provides small improvements on unknown tokens).

2.2 Domain Adaptation via word clustering

In our approach to user-generated content parsing, we use the domain adaptation method proposed by Candito et al. (2011). It consists in the following

[3] On the raw training corpora, the lowercasing strategy affects approximately 0.5% of all tokens, and both other correction strategies affect together 0.5% of all tokens as well.

	Ontonotes dev		e-mail dev		weblog dev	
	all	unk	all	unk	all	unk
MElt−corr	96.5	92.3	88.9	62.4	94.7	87.2
MElt+corr	96.5	92.9	90.4	72.1	94.7	87.3

Table 1: Evaluation results for the MElt POS-tagger, embedded or not within the normalization and correction wrapper ("MElt+corr" and "MElt−corr" respectively). Results are given on all tokens as well as on unknown tokens only ("all" and "unk" respectively.)

steps: (i) compute unsupervised word clusters over a mixture of raw corpora both from source and target domains, (ii) train a parser on a training corpus where each token is replaced by its (unambiguous) cluster (iii) use the same preprocessing for the text to parse, and parse it and (iv) reintroduce the original tokens in the resulting predicted parses. This method builds on the work of (Koo et al., 2008), in which unsupervised word clusters are used as features in a discriminative dependency parser, and of (Candito and Crabbé, 2009; Candito and Seddah, 2010) who proposed to use clusters as word substitutes in a generative constituency parser, and proved it useful for source domain parsing, by reducing lexical data sparseness. The use of clusters computed over source and target domains further helps to bridge the lexical gap between both domains, as some clusters group together source- and target-domain words.

In the case of the SANCL shared task, the target domain is spread over 5 subdomains of user generated content of inequal size (70 million tokens overall). The main difficulty of parsing such different domains is to handle different lexical data with potentially different distribution and of course to cope with a high level of out of vocabulary words. We propose to use the afore-mentioned lexical bridge technique to alleviate both issues.

2.3 Self-training architecture

On top of this word clustering technique, we use self-training, both as a standard technique for domain adaptation and to cope with a specific problem: the Ontonotes training set is unaware of a few POS tags attested in the development data (e.g., ADD), that are thus expected to appear in the test data. We proceed as follows:

1. **Baseline parser**: we train a baseline word-clustered parser on the Ontonotes training set using a 5 split-merge cycle and 4 grammars ;

2. **Bootstrapped parser**: we apply the tagger described in Section 2.1 on the raw in-domain corpora. We then randomly select 50 tagged sentences for each unknown or rare tag,[4] among the sentences that have a length between 7 and 20 words, and contain only one (word, tag) pair unknown to the Ontonotes training set — i.e., apart from the word bearing the unknown or rare tag, all (word, tag) pairs are known. This selection criteria aims at minimizing the risk of adding erroneous and therefore noisy trees in the training data. We parse these tagged bootstrap sentences with the baseline clustered parser (in provided-tag mode); unknown POS tags are ignored by the parser, and are re-injected afterwards; together with the Ontonotes training set, the resulting parses are used for training the (word-clustered) bootstrapped parser, which is now aware of all POS tags;

3. **Self-trained parser**: we randomly select 70,000 sentences from the raw in-domain corpora, using the same selection constraints as in the previous step; we parse them with the bootstrapped clustered parser, and add the resulting parses in the training set for the parser. The resulting word-clustered self-trained parser is then used for the final experiments.

3 Experiment Settings

The PCFG-LA Parser For our parsing experiments, we use the LORG parser[5] of Attia et al. (2010), which is an implementation of the PCFG with Latent Annotations (PCFG-LA) algorithm of Petrov and Klein (2007). Our experiments are run using five or six split-merge cycles (henceforth S5 or S6) without any special configuration for handling unknown words (so-called GENERIC mode). We use a product of either four or eight grammars (henceforth N4 or N8). The threshold under which tokens are considered unknown is 2 (only true unknown and

[4] The unknown tags are ADD and GW. The rare tags are the ones appearing with a frequency below $1/50000$: AFX and NFP.

[5] In its December 2011 version.

hapaxes). In addition, for improving the parsers' robustness, we replaced gold POS tags in the training data with tags obtained using a 20-fold jack-kniffing.

The parser is run in a special mode where POS tags are provided for the (word, tag) pairs that are unknown to the Ontonotes training set: these include (i) unknown words and (ii) known words which never appear with such a predicted tag in the training set. This is similar in spirit to the unknown tag supplied mode present in the Bikel's parser (Bikel, 2002), although the latter covers only case (i). Note that in case of parse failure, the system will backup to a configuration without supplied tags.[6]

Clustered word forms We generate 1,000 word clusters from both the unlabeled and training data sets, for words appearing at least 100 times. Before training and parsing, each input token is replaced by the concatenation of its cluster id, its 3 letter suffix and a feature marking capitalization. This cluster configuration is referred to as K-RAW. Table 2 displays some properties of the clustered training sets: in the self-training configuration (100k sentences), only 20k cluster types are used. As expected, lexical data sparseness is extremely reduced.

Corpus	Onto dev. + bootstrap		Onto dev. + bootstrap + Self-training data
Token type	raw	*K-raw*	*K-raw*
vocab. size	36052	15073	19893
# occurences	0.7M	0.7M	1.66M
# sentences	30220	30220	99433

Table 2: Lexical Properties of Training Sets

Baseline: results for the bootstrapped parser Our first results on the shared task development set (mail, blogs and Ontonotes) are presented Table 3 and show that our strategy brings a slight improvement over what is indeed a very strong baseline.

Final Results and Conclusion Based on the hypothesis that increasing the amount of training material would help to counteract any overfitting coming from an increase in the number of split-merge cycles, we decided to train and submit results output using a self-training configuration and 6 split-merge cycles. We retained two architectures, using a product of either 4 or 8 grammars. Those two instances of our architecture were ranked #2 and #3 among all participants of this shared task. Interesting results indeed, but unfortunately, looking closely at the s5 configurations (which could not be submitted in time, it is clear that our first results suffered from a strong overfitting as shown by the difference of performance between the S5+N8 and the S6+N8 settings. The latter providing higher results in the WSJ domain but inferior in all other domains. The relatively short size of the self training material and its homogeneity (recall that we selected sentences with at most one unknown word) added to the use of word clusters (which alleviates almost totally the notion of unknown words) may explain why an overfitting is experimented that "early", whereas in the Huang et al. (2010) work a performance drop, with a split-merge cycle of 7, is reported at around 170k sentences.

In all cases, our methodology proved useful and many paths of improvement will be explored in our future work.

[6]This happens for the parsing of the blind domains A and B.

	ONTO	MAIL	WEBLOG
no pos supplied			
Bracketing FMeasure	89.96	80.05	84.71
Tagging accuracy	95.58	86.12	92.24
K-raw, no tag			
Bracketing FMeasure	90.09	80.61	**85.43**
Tagging accuracy	96.32	88.23	94.02
K-raw, predicted tag on unknown (word, tag) pairs only			
Bracketing FMeasure	**90.17**	**81.06**	85.36
Tagging accuracy	96.60	90.81	94.80

Table 3: Baseline results (S5+N4, no self-training)

	BKY b.line	Alpage (off.) S6		*DCU -P13*	Alpage (unoff.) S5	
		N4	N8		N4	N8
Rank		#3	#2	#1	(#2')	(#1')
A	*75.92*	80.52	**80.60**	*82.19*	81.37	**81.46**
B	*78.14*	83.67	**84.03**	*84.33*	83.84	**84.13**
C	*77.16*	81.52	**81.76**	*84.03*	82.55	**82.68**
avg.	*77.07*	81.90	**82.13**	*83.52*	82.34	**82.45**
D	*88.21*	**89.91**	89.87	*90.53*	89.60	89.74

Table 4: Self-training configuration: Shared Task Results (F1 scores). Our official results ("Alpage") concern the S6+N4 and S6+N8 settings. Results for additional (better) settings are also given. *(A: answer domain, B: newsgroups, C: reviews, D: WSJ.)*

Acknowledgments

We thank D.Hogan, J.Foster and J.Le Roux for making their parser available to us. This work is partly funded by the French Research Agency (EDyLex, grant number ANR-09-COORD-008).

References

Mohammed Attia, Jennifer Foster, Deirdre Hogan, Joseph Le Roux, Lamia Tounsi, and Josef van Genabith. 2010. Handling unknown words in statistical latent-variable parsing models for arabic, english and french. In *Proceedings of the NAACL/HLT Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)*, Los Angeles, CA.

Daniel M. Bikel. 2002. Design of a multi-lingual, parallel-processing statistical parsing engine. In *Proceedings of the second international conference on Human Language Technology Research*, pages 178–182. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Peter F. Brown, Vincent J. Della, Peter V. Desouza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Marie Candito and Benoît Crabbé. 2009. Improving generative statistical parsing with semi-supervised word clustering. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 138–141, Paris, France, October. Association for Computational Linguistics.

Marie Candito and Djamé Seddah. 2010. Parsing word clusters. In *Proceedings of the NAACL/HLT Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)*, Los Angeles, CA.

Marie Candito, Enrique Henestroza Anguiano, and Djamé Seddah. 2011. A word clustering approach to domain adaptation: Effective parsing of biomedical texts. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 37–42, Dublin, Ireland, October. Association for Computational Linguistics.

Pascal Denis and Benoît Sagot. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art pos tagging with less human effort. In *Proc. of PACLIC*, Hong Kong, China.

J. Foster, J. Wagner, D. Seddah, and J. Van Genabith. 2007. Adapting wsj-trained parsers to the british national corpus using in-domain self-training. In *Proceedings of the Tenth IWPT*, pages 33–35.

Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Joseph Le Roux, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011. From news to comment: Resources and benchmarks for parsing the language of web 2.0. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 893–901, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Jennifer Foster. 2010. "cba to check the spelling": Investigating parser performance on discussion forum posts. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 381–384, Los Angeles, California, June. Association for Computational Linguistics.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 167–202, Pittsburgh, USA.

Z. Huang, M. Harper, and S. Petrov. 2010. Self-training with products of latent variable grammars. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 12–22. Association for Computational Linguistics.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08*, pages 595–603, Columbus, USA.

M. Lease and E. Charniak. 2005. Parsing biomedical literature. *Natural Language Processing–IJCNLP 2005*, pages 58–69.

G. Leech. 1992. 100 million words of english: the british national corpus. *Language Research*, 28(1):1–13.

Percy Liang. 2005. Semi-supervised learning for natural language. In *MIT Master's thesis*, Cambridge, USA.

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *Proceedings of ACL-08: HLT, Short Papers*, pages 101–104, Columbus, Ohio, June. Association for Computational Linguistics.

D. McClosky, E. Charniak, and M. Johnson. 2006a. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 337–344. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006b. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA, June. Association for Computational Linguistics.

D. McClosky, E. Charniak, and M. Johnson. 2010. Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of*

the North American Chapter of the Association for Computational Linguistics, pages 28–36. Association for Computational Linguistics.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, Rochester, New York, April. Association for Computational Linguistics.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

Benoît Sagot and Pierre Boullier. 2008. SXPipe 2 : architecture pour le traitement présyntaxique de corpus bruts. *Traitement Automatique des Langues (T.A.L.)*, 49(2).

Benoît Sagot. 2010. The Le*fff*, a freely available and large-coverage morphological and syntactic lexicon for french. In *Proceedings of LREC'10*, Valetta, Malta.

J. Wagner, D. Seddah, J. Foster, J. Van Genabith, M. Butt, and T.H. King. 2007. C-structures and f-structures for the british national corpus. In *Proceedings of the Twelfth International Lexical Functional Grammar Conference*. Citeseer.