The LIGM-Alpage Architecture for the SPMRL 2013 Shared Task: Multiword Expression Analysis and Dependency Parsing

Matthieu Constant
Université Paris-Est
LIGM
CNRS

Marie Candito
Alpage
Paris Diderot Univ
INRIA

Djamé Seddah
Alpage
Paris Sorbonne Univ
INRIA

Abstract

This paper describes the LIGM-Alpage system for the SPMRL 2013 Shared Task. We only participated to the French part of the dependency parsing track, focusing on the realistic setting where the system is informed neither with gold tagging and morphology nor (more importantly) with gold grouping of tokens into multi-word expressions (MWEs). While the realistic scenario of predicting both MWEs and syntax has already been investigated for constituency parsing, the SPMRL 2013 shared task datasets offer the possibility to investigate it in the dependency framework. We obtain the best results for French, both for overall parsing and for MWE recognition, using a reparsing architecture that combines several parsers, with both pipeline architecture (MWE recognition followed by parsing), and joint architecture (MWE recognition performed by the parser).

1 Introduction

As shown by the remarkable permanence over the years of specialized workshops, multiword expressions (MWEs) identification is still receiving considerable attention. For some languages, such as Arabic, French, English, or German, a large quantity of MWE resources have been generated (Baldwin and Nam, 2010). Yet, while special treatment of complex lexical units, such as MWEs, has been shown to boost performance in tasks such as machine translation (Pal et al., 2011), there has been relatively little work exploiting MWE recognition to improve parsing performance.

Indeed, a classical parsing scenario is to pregroup MWEs using gold MWE annotation (Arun and Keller, 2005). This non-realistic scenario has been shown to help parsing (Nivre and Nilsson, 2004; Eryigit et al., 2011), but the situation is quite different when switching to automatic MWE prediction. In that case, errors in MWE recognition alleviate their positive effect on parsing performance (Constant et al., 2012). While the realistic scenario of syntactic parsing with automatic MWE recognition (either done jointly or in a pipeline) has already been investigated in constituency parsing (Cafferkey et al., 2007; Green et al., 2011; Constant et al., 2012; Green et al., 2013), the French dataset of the SPMRL 2013 Shared Task (Seddah et al., 2013) offers one of the first opportunities to evaluate this scenario within the framework of dependency syntax.

In this paper, we discuss the systems we submitted to the SPMRL 2013 shared task. We focused our participation on the French dependency parsing track using the predicted morphology scenario, because it is the only data set that massively contains MWEs. Our best system ranked first on that track (for all training set sizes). It is a reparsing system that makes use of predicted parses obtained both with pipeline and joint architectures. We applied it to the French data set only, as we focused on MWE analysis for dependency parsing. Section 2 gives its general description, section 3 describes the handling of MWEs. We detail the underlying parsers in section 4 and their combination in section 5. Experiments are described and discussed in sections 6 and 7.

2 System Overview

Our whole system is made of several single statistical dependency parsing systems whose outputs are combined into a *reparser*. We use two types of sin-

Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically Rich Languages, pages 46–52,
Seattle, Washington, USA, 18 October 2013. ©2013 Association for Computational Linguistics

gle parsing architecture: (a) pipeline systems; (b) ”joint” systems.

The pipeline systems first perform MWE analysis before parsing. The MWE analyzer (section 3) merges recognized MWEs into single tokens and the parser is then applied on the sentences with this new tokenization. The parsing model is learned on a gold training set where all marked MWEs have been merged into single tokens. For evaluation, the merged MWEs appearing in the resulting parses are expanded, so that the tokens are exactly the same in gold and predicted parses.

The ”joint” systems directly output dependency trees whose structure comply with the French dataset annotation scheme. As shown in Figure 1, such trees contain not only syntactic dependencies, but also the grouping of tokens into MWEs, since the first component of an MWE bears dependencies to the subsequent components of the MWE with a specific label `dep_cpd`. At that stage, the only missing information is the POS of the MWEs, which we predict by applying a MWE tagger in a post-processing step.

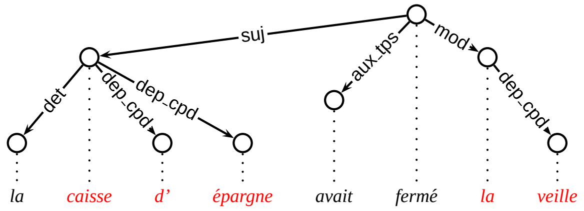

Figure 1: French dependency tree for *La caisse d'épargne avait fermé la veille* (*The savings bank had closed the day before*), containing two MWEs (in red).

3 MWE Analyzer and MWE Tagger

The MWE analyzer we used in the pipeline systems is based on Conditional Random Fields (CRF) (Lafferty et al., 2001) and on external lexicons following (Constant and Tellier, 2012). Given a tokenized text, it jointly performs MWE segmentation and POS tagging (of simple tokens and of MWEs), both tasks mutually helping each other[1]. CRF is a prominent statistical model for sequence segmentation and labelling. External lexicons used as sources of features greatly improve POS tagging (Denis and Sagot, 2009) and MWE segmentation (Constant and Tellier, 2012). Our lexical resources are composed of two large-coverage general-language lexicons: the Lefff[2] lexicon (Sagot, 2010), which contains approx. half a million inflected word forms, among which approx. $25,000$ are MWEs; and the DELA[3] (Courtois, 2009; Courtois et al., 1997) lexicon, which contains approx. one million inflected forms, among which about $110,000$ are MWEs. These resources are completed with specific lexicons freely available in the platform Unitex[4]: the toponym dictionary Prolex (Piton et al., 1999) and a dictionary of first names.

The MWE tagger we used in the joint systems takes as input a MWE within a dependency tree, and outputs its POS. It is a pointwise classifier, based on a MaxEnt model that integrates different features capturing the MWE local syntactic context, and in particular the POS at the token level (and not at the MWE level). The features comprise: the MWE form, its lemma, the sequence of POS of its components, the POS of its first component, its governor's POS in the syntactic parse, the POS following the MWE, the POS preceding the MWE, the bigram of the POS following and preceding the MWE.

4 Dependency Parsers

For our development, we trained 3 types of parsers, both for the pipeline and the joint architecture:

- **MALT**, a pure linear-complexity transition-based parser (Nivre et al., 2006)
- **Mate-tools 1**, the graph-based parser available in *Mate-tools*[5] (Bohnet, 2010)
- **Mate-tools 2**, the joint POS tagger and transition-based parser with graph-based completion available in *Mate-tools* (Bohnet and Nivre, 2012).

[1] Note though that we keep only the MWE segmentation, and use rather the Morfette tagger-lemmatizer, cf. section 4.

[2] We use the version available in the POS tagger MElt (Denis and Sagot, 2009).

[3] We use the version in the platform Unitex (http://igm.univ-mlv.fr/˜unitex). We had to convert the DELA POS tagset to the FTB one.

[4] http://igm.univ-mlv.fr/˜unitex

[5] Available at http://code.google.com/p/mate-tools/. We used the *Anna3.3* version.

Such parsers require some preprocessing of the input text: lemmatization, POS tagging, morphology analyzer (except the joint POS tagger and transition-based parser that does not require preprocessed POS tagging). We competed for the scenario in which this information is not gold but predicted. Instead of using the predicted POS, lemma and morphological features provided by the shared task organizers, we decided to retrain the tagger-lemmatizer Morfette (Chrupała et al., 2008; Seddah et al., 2010), in order to apply a jackknifing on the training set, so that parsers are made less sensitive to tagging errors. Note that no feature pertaining to MWEs are used at this stage.

5 Reparser

The reparser is an adaptation to labeled dependency parsing of the simplest[6] system proposed in (Sagae and Lavie, 2006). The principle is to build an arc-factored merge of the parses produced by n input parsers, and then to find the maximum spanning tree among the resulting merged graph[7]. We implemented the maximum spanning tree algorithm[8] of (Eisner, 1996) devoted to projective dependency parsing. During the parse merging, each arc is unlabeled, and is given a weight, which is the frequency it appears in the n input parses. Once the maximum spanning tree is found, each arc is labeled by its most voted label among the m input parses containing such an arc (with arbitrary choice in case of ties).

6 Experiments

6.1 Settings

MWE Analysis and Tagging

For the MWE analyzer, we used the tool *lgtagger*[9] (version 1.1) with its default set of feature templates. The MWE tagger model was trained using the Wapiti software(Lavergne et al., 2010). We used the default parameters and we forced the MaxEnt mode.

Parsers

For MALT (version 1.7.2), we used the *arceager* algorithm, and the *liblinear* library for training. As far as the features are concerned, we started with the feature templates given in Bonsai[10] (Candito et al., 2010), and we added some templates (essentially lemma bigrams) during the development tests, that slightly improved performance. For the two *Mate-tools* parsers, we used the default feature sets and parameters proposed in the documentation.

Morphological prediction

Predicted lemmas, POS and morphology features are computed with Morfette version 0.3.5 (Chrupała et al., 2008; Seddah et al., 2010)[11], using 10 iterations for the tagging perceptron, 3 iterations for the lemmatization perceptron, default beam size for the decoding of the joint prediction, and the Lefff (Sagot, 2010) as external lexicon used for out-of-vocabulary words. We performed a jackknifing on the training corpus, with 10 folds for the full corpus, and 20 folds for the 5k track[12].

6.2 Results

We first provide the results on the development corpus. Table 1 shows the general parsing accuracy of our different systems. Results are displayed in three different groups corresponding to each kind of systems: the two single parser architectures ones (joint and pipeline) and the reparsing one. Each system was tested both when learned on the full training data set and on the 5k one. The joint and pipeline systems were evaluated with the three parsers described in section 4. For the reparser, we tested different combinations of parsers in the full training data set mode. We found that the best combination includes all parsers but MALT in joint mode. We did not tune our reparsing system in the 5k training data set mode. We assumed that the best combination in this mode was the same as with full training.

[6] The other more complex systems were producing equivalent scores.

[7] In order to account for labeled MWE recognition, we integrated in the ”dep_cpd” arcs the POS of the corresponding MWE. For instance, if the label ”dep_cpd” corresponds to an arc in a multiword preposition (P), the arc is relabeled ”dep_cpd_P”. At evaluation time, the output parse labels are remapped to the official annotation scheme.

[8] More precisely, we based our implementation on the pseudo-code given in (McDonald, 2006).

[9] http://igm.univ-mlv.fr/˜mconstan

[10] http://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html

[11] Available at https://sites.google.com/site/morfetteweb/

[12] Note that for the 5k track, we retrained Morfette using the 5k training corpus only, whereas the official 5k training set contains predicted morphology trained on the full training set.

		full			5k		
type	parser	LAS	UAS	LaS	LAS	UAS	LaS
Joint	MALT	80.91	84.74	89.18	78.61	83.16	87.51
	Mate-tools 1	84.60	88.21	91.43	82.02	86.23	90.02
	Mate-tools 2	84.40	88.08	91.02	81.66	85.97	89.38
Pipeline	MALT	82.56	86.22	90.22	80.79	84.71	89.19
	Mate-tools 1	85.28	88.73	91.85	83.23	86.97	90.67
	Mate-tools 2	84.82	88.31	91.45	82.79	86.56	90.26
Reparser	joint only	85.28	88.77	91.70	-	-	-
	pipeline only	85.79	89.17	91.94	-	-	-
	all	86.12	89.36	92.22	-	-	-
	best ensemble	86.23	89.55	92.21	84.25	87.88	91.17

Table 1: Parsing results on development corpus (38820 tokens)

	COMP			MWE			MWE+POS		
	R	P	F	R	P	F	R	P	F
joint Mate-tools 1	76.3	82.4	79.2	74.3	80.6	77.3	70.7	76.7	73.6
pipeline Mate-tools 1	80.8	82.7	81.7	79.0	83.6	81.2	75.6	80.1	77.8
best reparser	81.1	82.5	81.8	79.2	83.0	81.0	76.1	79.8	77.9

Table 2: MWE Results on the development corpus (2119 MWEs) with full training.

Table 2 contains the MWE results on the development data set with full training, for three systems: the best single-parser joint and pipeline systems (i.e. with Mate-tools 1) and the best reparser. We do not provide results for the 5k training because they show similar trends. We provide the 9 MWE-related measures defined in the shared task. The symbols *R*, *P* and *F* respectively correspond to *recall*, *precision* and *F-measure*. *COMP* corresponds to evaluation of the non-head MWE components (i.e. the non-first MWE components, cf. Figure 1). *MWE* corresponds to the recognition of a complete MWE. *MWE+POS* stands for the recognition of a complete MWE associated with its correct POS.

We submitted to the shared task our best (reparser) system according to the tuning described above. We also sent the two best pipeline systems (Mate-tools 1 and Mate-tools 2) and the best joint system (Mate-tools 1), in order to compare our single systems to the other competitors. The official results of our systems are provided in table 3 for general parsing and in table 4 for MWE recognition. We also show the ranking of each of these systems in the competition.

7 Discussion

In table 3, we can note that for the $5k$ training set scenario, there is a general drop of parsing performance (approximately 2 points), but the trends are exactly the same as for the full training set scenario. Concerning the performance on MWE analysis (table 4), the pipeline Mate-tools-1 system very slightly outperforms the best reparser system in the 5k scenario, contrary to the full training set scenario, but the difference is not significant. In the following, we focus on the full training set scenario.

Let us first discuss the overall parsing performance, by looking at the results on the development corpus (table 1). As far as the single-parser systems are concerned, we can note that for both the joint and pipeline systems, MALT achieves lower performance than the graph-based (Mate-tools-1) and the joint tagger-parser (Mate-tools-2), which have comparable performance. Moreover, the pipeline systems achieve overall better than their joint counterpart, though the increase between joint and pipeline architecture is much bigger for MALT than for the Mate parsers (for MALT, compare

training	type	parser	LAS	UAS	LaS	Rank
Full	Reparser	best	85.86	89.19	92.20	1
	Pipeline	Mate-tools 1	84.91	88.35	91.73	3
	Pipeline	Mate-tools 2	84.87	88.40	91.51	4
	Joint	Mate-tools 1	84.14	87.67	91.24	7
5k	Reparser	best	83.60	87.40	90.76	1
	Pipeline	Mate-tools 1	82.53	86.51	90.14	4
	Pipeline	Mate-tools 2	82.15	86.18	89.79	6
	Joint	Mate-tools 1	81.63	85.76	89.56	7

Table 3: Official parsing results on the evaluation corpus (75216 tokens)

training	type	parser	COMP	MWE	MWE+POS	Rank
Full	Reparser	best ensemble	81.3	80.7	77.5	1
	Pipeline	Mate-tools 1	81.2	80.8	77.4	2
	Pipeline	Mate-tools 2	81.2	80.8	76.6	3
	Joint	Mate-tools 1	79.6	77.4	74.1	6
5k	Pipeline	Mate-tools 1	78.7	77.7	74.0	1
	Reparser	best ensemble	78.9	77.2	73.8	2
	Pipeline	Mate-tools 2	78.7	77.7	73.3	5
	Joint	Mate-tools 1	75.9	72.2	75.9	10

Table 4: Official MWE results on the evaluation corpus (4043 MWEs). The scores correspond to the F-measure.

LAS=80.91 for the joint system, and LAS=82.56 for the pipeline architecture, while for Mate-tools-1, compare LAS=84.60 with LAS=85.28). The best reparser system provides a performance increase of approximately one point over the best single-parser system (Mate-tools-1), both for LAS and UAS, which suggests that the parsers have complementary strengths.

When looking at performance on MWE recognition and tagging (2), we can note greater variation between the F-measures obtained by the single-parser systems, but this is due to the much lower number of MWEs with respect to the number of tokens (there are 38820 tokens and 2119 MWEs in the dev set). The MWE analyzer used in the pipeline systems leads to better MWE recognition ($F - measure = 81.2$ on dev set) than when the analysis is left to the bare “joint” parsers (joint Mate-tools 1 achieves F-measure= 77.3).

Contrary to the situation for overall parsing performance, the reparser system does not lead to better MWE recognition with respect to the MWE analyzer of the pipeline systems. Indeed the performance on MWEs are quite similar between the reparser system and the MWE analyzer (for the MWE metric, on the dev set we get F=81.0 versus 81.2 for best reparser and pipeline systems respectively, whereas we get 80.7 and 80.8 on the test set. These differences are not significant). This is because the MWEs predicted by the MWE analyzer are present in three of the single-parser systems taken into account in the reparsing process, and are thus much favored in the voting.

In order to understand better our parsing systems' performance on MWE recognition, we provide in table 5 the MWE+POS results broken down by MWE part-of-speech, for the dev set. Not surprisingly, we can note that performance varies greatly depending on the POS, with better performance on closed classes (conjunctions, determiners, prepositions, pronouns) than on open classes. The lowest performance is on adjectives and verbs, but given the raw numbers of gold MWEs, the major impact on overall performance is given by the results on nominal MWEs (either common or proper nouns). A little less than one third of the nominal gold MWEs

	R	P	F	Nb gold	Nb predicted	Nb correct
adjectives	46.9	75.0	57.7	32	20	15
adverbs	74.7	83.0	78.7	360	324	269
conjunctions	90.1	83.7	86.8	91	98	82
clitics	-	0.00	-	0	1	0
determiners	96.0	96.8	96.4	252	250	242
nouns	72.7	76.2	74.4	973	928	707
prepositions	84.6	84.9	84.8	345	344	292
pronouns	75.0	87.5	80.8	28	24	21
verbs	66.7	66.7	66.67	33	33	22
unknown	0	0	0	5	0	0
ALL	77.9	81.6	79.7	2119	2022	1650

Table 5: MWE+POS results on the development corpus, broken down by POS (recall, precision, F-measure, number of gold MWEs, predicted MWEs, correct MWEs with such POS.

is not recognized (R = 72.7), and about one quarter of the predicted nominal MWEs are wrong (P = 76.2). Though these results can be partly explained by some inconsistencies in MWE annotation in the French Treebank (Constant et al., 2012), there remains room for improvement for open class MWE recognition.

8 Conclusion

We have described the LIGM-Alpage system for the SPMRL 2013 shared task, restricted to the French track. We provide the best results for the realistic scenario of predicting both MWEs and dependency syntax, using a reparsing architecture that combines several parsers, both pipeline (MWE recognition followed by parsing) and joint (MWE recognition performed by the parser). In the future, we plan to integrate features specific to MWEs into the joint system, so that the reparser outperforms both the joint and pipeline systems, not only on parsing (as it is currently the case) but also on MWE recognition.

References

A. Arun and F. Keller. 2005. Lexicalization in crosslinguistic probabilistic parsing: The case of french. In *Proceedings of the Annual Meeting of the Association For Computational Linguistics (ACL'05)*, pages 306–313.

T. Baldwin and K.S. Nam. 2010. Multiword expressions. In *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1455–1465, Jeju Island, Korea, July. Association for Computational Linguistics.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*, Beijing, China.

C. Cafferkey, D. Hogan, and J. van Genabith. 2007. Multi-word units in treebank-based probabilistic parsing and generation. In *Proceedings of the 10th International Conference on Recent Advances in Natural Language Processing (RANLP'07)*.

M.-H. Candito, J. Nivre, P. Denis, and E. Henestroza Anguiano. 2010. Benchmarking of statistical dependency parsers for french. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*, Beijing, China.

Grzegorz Chrupała, Georgiana Dinu, and Josef van Genabith. 2008. Learning morphology with morfette. In *In Proc. of LREC 2008*, Marrakech, Morocco. ELDA/ELRA.

Matthieu Constant and Isabelle Tellier. 2012. Evaluating the impact of external lexical resources into a crf-based multiword segmenter and part-of-speech tagger. In *Proceedings of the 8th conference on Language Resources and Evaluation (LREC'12)*.

Matthieu Constant, Anthony Sigogne, and Patrick Watrin. 2012. Discriminative strategies to integrate multiword expression recognition and parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 204–212, Stroudsburg, PA, USA. Association for Computational Linguistics.

B. Courtois, M. Garrigues, G. Gross, M. Gross, R. Jung, M. Mathieu-Colas, A. Monceaux, A. Poncet-Montange, M. Silberztein, and R. Vivés. 1997. Dictionnaire électronique DELAC : les mots composés binaires. Technical Report 56, University Paris 7, LADL.

B. Courtois. 2009. Un système de dictionnaires électroniques pour les mots simples du français. *Langue Française*, 87:11–22.

P. Denis and B. Sagot. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC'09)*, pages 110–119.

Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: an exploration. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, COLING '96, pages 340–345, Stroudsburg, PA, USA. Association for Computational Linguistics.

G. Eryigit, T. Ilbay, and O. Arkan Can. 2011. Multiword expressions in statistical dependency parsing. In *Proceedings of the IWPT Workshop on Statistical Parsing of Morphologically-Rich Languages (SPRML'11)*, pages 45–55.

S. Green, M.-C. de Marneffe, J. Bauer, and C. D. Manning. 2011. Multiword expression identification with tree substitution grammars: A parsing tour de force with french. In *Proceedings of the conference on Empirical Method for Natural Language Processing (EMNLP'11)*, pages 725–735.

Spence Green, Marie-Catherine de Marneffe, and Christopher D Manning. 2013. Parsing models for identifying multiword expressions. *Computational Linguistics*, 39(1):195–227.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML'01)*, pages 282–289.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, pages 504–513.

Ryan McDonald. 2006. *Discriminative Training and Spanning Tree Algorithms for Dependency Parsing*. Ph.D. thesis, University of Pennsylvania.

J. Nivre and J. Nilsson. 2004. Multiword units in syntactic parsing. In *Proceedings of Methodologies and Evaluation of Multiword Units in Real-World Applications (MEMURA)*.

J. Nivre, J. Hall, and J. Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC'06)*, pages 2216–2219, Genoa, Italy.

Santanu Pal, Tanmoy Chkraborty, and Sivaji Bandyopadhyay. 2011. Handling multiword expressions in phrase-based statistical machine translation. In *Proceedings of the Machine Translation Summit XIII*, pages 215–224.

O. Piton, D. Maurel, and C. Belleil. 1999. The prolex data base : Toponyms and gentiles for nlp. In *Proceedings of the Third International Workshop on Applications of Natural Language to Data Bases (NLDB'99)*, pages 233–237.

Kenji Sagae and Alon Lavie. 2006. Parser combination by reparsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 129–132, Stroudsburg, PA, USA. Association for Computational Linguistics.

B. Sagot. 2010. The lefff, a freely available, accurate and large-coverage lexicon for french. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10)*.

Djamé Seddah, Grzegorz Chrupała, Ozlem Cetinoglu, Josef van Genabith, and Marie Candito. 2010. Lemmatization and statistical lexicalized parsing of morphologically-rich languages. In *Proc. of the NAACL/HLT Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)*, Los Angeles, CA.

Djamé Seddah, Reut Tsarfaty, Sandra Ḱubler, Marie Candito, Jinho Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiorkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clérgerie. 2013. Overview of the spmrl 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the 4th Workshop on Statistical Parsing of Morphologically Rich Languages: Shared Task*, Seattle, WA.