IXA pipeline: Efficient and Ready to Use Multilingual NLP tools

Rodrigo Agerri, Josu Bermudez, German Rigau

rodrigo.agerri@ehu.es, josu.bermudez@deusto.es, german.rigau@ehu.es
*IXA NLP Group, University of the Basque Country (UPV/EHU), Donostia-San Sebastián
**Deusto Institute of Technology - DeustoTech, University of Deusto, Bilbao

Abstract

IXA pipeline is a modular set of Natural Language Processing tools (or pipes) which provide easy access to NLP technology. It offers robust and efficient linguistic annotation to both researchers and non-NLP experts with the aim of lowering the barriers of using NLP technology either for research purposes or for small industrial developers and SMEs. IXA pipeline can be used "as is" or exploit its modularity to pick and change different components. Given its open-source nature, it can also be modified and extended for it to work with other languages. This paper describes the general data-centric architecture of IXA pipeline and presents competitive results in several NLP annotations for English and Spanish.

1. Introduction

Many Natural Language Processing (NLP) applications demand some basic linguistic processing (Tokenization, Part of Speech (POS) tagging, Named Entity Recognition and Classification (NER), constituent parsing, Coreference Resolution, etc.) to be able to further undertake more complex tasks. Generally, NLP annotation is required to be as accurate and efficient as possible and existing tools, quite righly, have mostly focused on performance. However, this generally means that NLP suites and tools usually require researchers to do complex compilation/installation/configuration in order to use such tools.
At the same time, in the industry, there are currently many Small and Medium Enterprises (SMEs) offering services that one way or another depend on NLP annotations.
In both cases, in research and industry, acquiring, deploying or developing such base qualifying technologies is an expensive undertaking that redirects their original central focus: In research, much time is spent in the preliminaries of a particular research experiment trying to obtain the required basic linguistic annotation, whereas in an industrial environment SMEs see their already limited resources taken away from offering products and services that the market demands.
IXA pipeline provides *ready to use modules* to perform efficient and accurate linguistic annotation. More specifically, the objectives of IXA pipeline is to offer basic NLP technology that is:

1. **Simple and ready to use**: Every module of the IXA pipeline can be up an running after two simple steps.

2. **Portable**: The binaries are generated with "all batteries included" which means that it is not required to do any system configuration or install any third-party dependencies. The modules will run on any platform as long as a JVM 1.7+ and/or Python 2.7 are available.

3. **Modular**: Unlike other NLP toolkits, which often are built in a monolithic architecture, IXA pipeline is built in a data centric architecture so that modules can be picked and changed (even from other NLP toolkits). The modules behave like Unix pipes, they all take standard input, do some annotation, and produce standard output which in turn is the input for the next module. The data-centric arquitecture of IXA pipeline makes every module *independent* and can therefore be used with other tools from other toolkits if required.

4. **Efficient**: Piping the tokenizer (250K per second), POS tagger and lemmatizer all in one process annotates over 5500 words/second. The Named Entity Recognition and Classification (NERC) module annotates over 5K words/second. In a multi-core machine, these times are dramatically reduced due to multithreading (even 4 times faster).

5. **Multilingual**: Currently we offer NLP annotations for both English and Spanish, but other languages are currently being included in the pipeline.

6. **Accurate**: Previous points do not mean that IXA pipeline does not strive to offer accurate linguistic annotators. For example, POS tagging and NERC for English and Spanish are comparable with other state of the art systems, as it is the coreference resolution module for English.

7. **Apache License 2.0**: IXA Pipeline is licensed under the Apache License 2.0, an open-source license that facilitates source code use, distribution and integration, also for commercial purposes.[1]

In the next section we describe the *IXA pipeline* architecture and in section 3. the modules that are already publicly available. Whenever available, we also present empirical evaluation. In section 4. we compare with some previous work and finally, in section 5. we conclude and present future ongoing work.

2. Architecture

IXA pipeline is primarily conceived as a set of *ready to use tools* that can provide efficient and accurate linguistic annotation without any installation/configuration/compilation effort. As in Unix-like operative systems, IXA pipeline

[1] http://www.apache.org/licenses/LICENSE-2.0.html

consists of a set of processes chained by their standard streams, in a way that the output of each process feeds directly as input to the next one. The Unix pipeline metaphor has been applied for NLP tools by adopting a very simple and well known data centric architecture, in which every module/pipe is interchangeable by any other tool as long as it reads and writes the required data format via the standard streams. Figure 1 shows the modules currently included in the IXA pipeline and the data-centric architecture with modules that interact by piping their standard streams.

The data format in which both the input and output of the modules needs to be formatted to represent and pipe linguistic annotations is KAF (Bosma et al., 2009). KAF is a language neutral annotation format representing both morpho-syntactic and semantic annotation in a structured format. KAF was originally designed in the Kyoto European project[2], but it has since been in continuous development[3] and evolved into NAF[4]. Our Java modules all use the kaflib[5] library for easy integration.

Every module in the IXA pipeline, except the coreference resolution, is implemented in Java, and requires Java JDK1.7+ to compile. The tools will work with Java 1.6, but without Unicode support for regular expressions, which is required for tokenization. The integration of the modules in the IXA pipeline is facilitated by means of Maven[6]. Maven is used to automatically perform any classpaths configurations and to instally any third-party tool dependencies. This means that the generated binaries and distributed *will work off-the-self*. Moreover, if the source code is cloned from the remote repository, *one command* to compile and have ready the tools will suffice.

Some modules in the IXA pipeline provide linguistic annotation based on probabilistic supervised approaches such as POS tagging, NERC and constituent parsing. IXA pipeline uses two well known machine learning algorithms, namely, Maximum Entropy and the Perceptron. Both Perceptron (Collins, 2002; Collins, 2003; Ratinov and Roth, 2009) and Maximum Entropy models (Ratnaparkhi and others, 1996; Charniak, 2000; Clark and Curran, 2003; Toutanova et al., 2003; Nothman et al., 2012) are adaptable algorithms which have largely been applied to NLP tasks such as POS tagging, NERC and Parsing with state of the art results. To avoid duplication of efforts, IXA pipeline uses the open-source Apache OpenNLP API[7] to train probabilistic models using these two approaches:

- A *perceptron* algorithm, where each outcome is represented as a binary perceptron classifier. Its implementation is based on Collins (2002).
- *Maximum Entropy* models based on Ratnaparkhi's approach (1996; 1998) optimized using Generalised Iterative Scaling.

[2]http://kyoto-project.eu
[3]http://www.opener-project.org/kaf/
[4]http://www.newsreader-project.eu/results/formats
[5]https://github.com/ixa-ehu/kaflib
[6]http://maven.apache.org/
[7]http://opennlp.apache.org

3. Pipes

IXA pipeline currently provides the following linguistic annotations: Sentence segmentation, tokenization, Part of Speech (POS) tagging, lemmatization, Named Entity Recognition and Classification (NERC), constituent parsing and coreference resolution. Every module works for English and Spanish and is implemented in Java/Maven as described above. The only exception is the coreference resolution module, which currently is available in Python 2.7 and for English only (Spanish version coming soon). The coreference module is also developed as an easy and ready to use tool, requiring only two steps to get it working. We will now describe which annotations are provided by each module of the pipeline.

3.1. ixa-pipe-tok

This module provides Sentence Segmentation and Tokenization for English and Spanish and other languages such as French, Italian, etc. It is the only module in the pipeline that takes plain text as input. It produces tokenized and segmented text in KAF, running text and CoNLL formats. It implements a rule-based segmenter and tokenizer originally inspired by the Stanford English Penn Treebank tokenizer[8] but with several modifications and improvements. These include tokenization for other languages such as French and Italian, normalization according the Spanish Ancora Corpus (Taulé et al., 2008), paragraph treatment, and more comprehensive gazeteers of non breaking prefixes. The tokenizer depends on a JFlex[9] specification file which compiles in seconds and performs at a very reasonable speed (around 250K word/second, and much quicker with Java multithreading).

3.2. ixa-pipe-pos

ixa-pipe-pos provides POS tagging and lemmatization for English and Spanish. We have obtained the best results so far with the following two models and the same featureset as in Collins's (2002) paper:

- *Perceptron* models for English have been trained and evaluated on the WSJ treebank using the usual partitions (e.g., as explained in Toutanova et al. (2003)). We currently obtain a performance of 96.88% vs 97.24% in word accuracy obtained by Toutanova et al., (2003).
- *Maximum Entropy* models have been trained and evaluated for Spanish using the Ancora corpus; it was randomly divided in 90% for training and 10% for testing. This corresponds to 440K words used for training and 70K words for testing. We obtain a performance of 98.88% (the corpus partitions are available for reproducibility). Giménez and Marquez (2004) report 98.86%, although they train and test on a different subset of the Ancora corpus.

Lemmatization is currently performed via 3 different dictionary lookup methods:

[8]http://www-nlp.stanford.edu/software/tokenizer.shtml
[9]http://jflex.de/

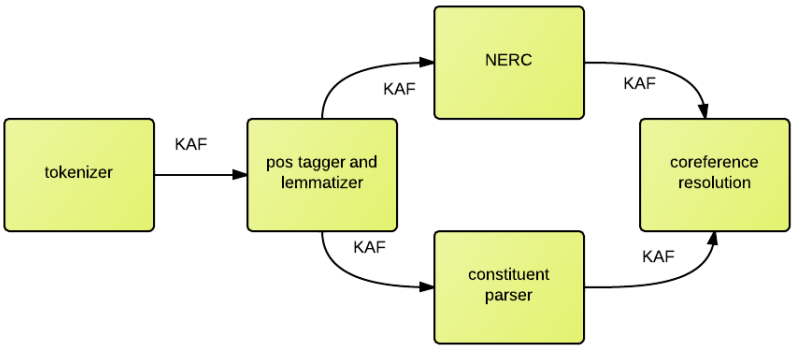

Figure 1: IXA pipeline data centric architecture.

1. *Simple Lemmatizer*: It is based on HashMap lookups on a plain text dictionary. Currently we use dictionaries from the LanguageTool project[10] under their distribution licenses. The English dictionary contains 300K lemmas whereas the Spanish provides over 600K.

2. *Morfologik-stemming*[11]: The Morfologik library provides routines to produce binary dictionaries, from dictionaries such as the one used by the Simple Lemmatizer above, as finite state automata. This method is convenient whenever lookups on very large dictionaries are required because it reduces the memory footprint to 10% of the memory required for the equivalent plain text dictionary.

3. We also provide lemmatization by lookup in *WordNet-3.0* (Fellbaum and Miller, 1998) via the JWNL API[12]. Note that this method is only available for English.

ixa-pipe-pos also allows to output the annotation in CoNLL style tabulated format whenever POS tagging and lemmatization are the last processes in the toolchain. Furthermore, we are currently working on improving both POS tagging and lemmatization results.

3.3. ixa-pipe-nerc

Most of the Named Entity Recognition and Classification (NERC) systems nowdays consist of language independent systems (sometimes enriched with gazeteers) based on automatic learning of statistical models (Nadeau and Sekine, 2007).

ixa-pipe-nerc provides NERC for English and Spanish. The named entity types are based on the CONLL 2002[13] and 2003[14] tasks which were focused on language-independent supervised named entity recognition for four types of named entities: persons, locations, organizations and names of miscellaneous entities that do not belong to the previous three groups. The participants of the shared tasks were offered training and test data for Dutch, Spanish (2002), English and German (2003) and the objective was to build NERC systems based on machine learning techniques. We currently provide two very fast language independent models using a rather simple baseline feature set. It is based on the features presented by Zhang and Johnson (2003) with several differences: We do not use POS tags, chunking or gazetteers in our baseline models but we do use bigrams as a feature.

- For English, perceptron models have been trained using the CoNLL 2003 dataset. We currenly obtain 84.70 F1 using also the development set for training (83.77 F1 without it). This is coherent with other results reported with these feature sets (Clark and Curran, 2003; Zhang and Johnson, 2003; Ratinov and Roth, 2009). The best Stanford NERC model reported on this dataset achieves 86.86 F1 (Finkel et al., 2005), whereas the best system on this dataset achieves 90.80 F1 (Ratinov and Roth, 2009), using non local features and substantial external knowledge.

- For Spanish we currently obtain best results training Maximum Entropy models on the CoNLL 2002 dataset. Our best model obtains 80.16 F1 vs 81.39 F1 of Carreras et al. (2002), the best result so far on this dataset. Their result uses external knowledge and without it, they system obtains 79.28 F1.

We are currently improving ixa-pipe-nerc in three directions:

1. Training models for different Named Entity types, e.g., using MUC-7 and Ontonotes 4.0 datasets (Weischedel et al., 2010) for English and Ancora corpus for Spanish (Taulé et al., 2008).

2. Adding new features to the baseline models here presented partially following previously published results (Finkel et al., 2005; Ratinov and Roth, 2009).

3. Representing named entities in the BILOU representation format as previous work shows that replacing BIO with the BILOU format improves performance (Ratinov and Roth, 2009).

[10] http://languagetool.org/
[11] https://github.com/morfologik/morfologik-stemming
[12] http://jwordnet.sourceforge.net/
[13] http://www.clips.ua.ac.be/conll2002/ner/
[14] http://www.clips.ua.ac.be/conll2003/ner/

As already mention for previous modules, *ixa-pipe-nerc* also allows to format its output in CoNLL style tabulated BIO format as specified in the CoNLL 2003 shared evaluation task. Note however that to pipe the output of *ixa-pipe-nerc* into another annotator it is still required to choose KAF/NAF as output format.

3.4. ixa-pipe-parse

ixa-pipe-parse provides statistical constituent parsing for English and Spanish. Maximum Entropy models are trained to build shift reduce bottom up parsers (Ratnaparkhi, 1999) as provided by the Apache OpenNLP Machine Learning API. Parsing models for English have been trained using the Penn treebank and for Spanish using the Ancora corpus (Taulé et al., 2008).

Furthermore, *ixa-pipe-parse* provides two methods of headword finders: one based on Collins' head rules as defined in his PhD thesis (1999), and another one based on Stanford's parser Semantic Head Rules[15]. The latter are a modification of Collins' head rules according to lexical and semantic criteria. These head rules are particularly useful for the Coreference resolution module. In addition to KAF/NAF output, *ixa-pipe-parse* also allows to output the parse trees into Penn Treebank bracketing style.

As far as we know, and although previous approaches exist (Cowan and Collins, 2005), *ixa-pipe-parse* provides the first publicly available statistical parser for Spanish.

3.5. Coreference Resolution

The module of coreference resolution included in the IXA pipeline is loosely based on the Stanford Multi Sieve Pass system (Lee et al., 2013). The module takes every linguistic information it requires from the KAF layers annotated by all the previously described modules.

The system consists of a number of rule-based sieves. Each sieve pass is applied in a deterministic manner, reusing the information generated by the previous sieve and the mention processing. The order in which the sieves are applied favours a highest precision approach and aims at improving the recall with the subsequent application of each of the sieve passes.

This is illustrated by the evaluation results of the CoNLL 2011 Coreference Evaluation task (Lee et al., 2011; Lee et al., 2013), in which the Stanford's system obtained the best results. The results show a pattern which has also been shown in other results reported with other evaluation sets (Raghunathan et al., 2010), namely, the fact that a large part of the performance of the multi pass sieve system is based on a set of significant sieves. Thus, this module focuses for the time being, on a **subset of sieves only**, namely, Speaker Match, Exact Match, Precise Constructs, Strict Head Match and Pronoun Match (Lee et al., 2013).

So far we have evaluated our module on the dev-auto part of the Ontonotes 4.0 corpus. Table 1 shows that we score around 3 CoNLL F1 points worse than Stanford's system (Lee et al., 2013). It is interesting that in our current implementation, mention-based metrics are favoured (CEAF and B^3). Still, note that these results are comparable with the results obtained by the best CoNLL 2011 participants.

[15]http://www-nlp.stanford.edu/software/lex-parser.shtml

The data-centric architecture of IXA pipeline means that our coreference resolution module, unlike most of the other coreference systems available, is *highly independent* and can therefore be used with other tools from other toolkits if required. Currently the module performs coreference resolution only for English, although a Spanish version will soon be available.

4. Related Work

Other NLP toolkits exist providing similar or more extensive functionalities than the IXA pipeline tools, although not many of them provide multilingual support: GATE (Cunningham et al., 2002) is an extensive framework supporting annotation of text. GATE has some capacity for wrapping Apache UIMA components[16], so should be able to manage distributed NLP components. However, GATE is a very large and complex system, with a corresponding steep learning curve.

Freeling (Padró and Stanilovsky, 2012) provides multilingual processing for a number of languages, including Spanish and English. As opposed to IXA pipeline, Freeling is a monolithic toolkit written in C++ which needs to be compiled natively. The Stanford CoreNLP[17] is a monolithic suite, which makes it difficult to integrate other tools in its chain.

The modules of IXA pipeline as shown by Figure 1 can easily be used piping the input with the output of another module via KAF. This makes it possible to easily replace or extend the toolchain with a third-party tool. The only requirement is for a tool to write and read KAF/NAF format. We have produced KAF/NAF wrappers for other modules, such as some of the Stanford CoreNLP tools. For example, we can easily change the parsing in our pipeline by replacing *ixa-pipe-parse* with a module wrapping the Stanford parser (Klein and Manning, 2003) to read and write KAF via standard output.

The IXA pipeline provides the, as fas as we know, only publicly available probabilistic parser for Spanish. Moreover, the IXA pipeline offers easy, ready to use and to replace NLP tools that do not require any effort for researchers or SMEs to get them up and running. Additionally, every tool provides evaluation functionalities following the usual conventions and metrics for each task. Finally it also offers the possibility of easily training new models with your own data for POS tagging, NERC and constituent parsing. The robustness of the IXA pipeline tools is already being tested doing extensive parallel processing in the FP7 European projects OpeNER[18] and NewsReader[19].

5. Conclusion and Future Work

IXA pipeline provides a simple, efficient, accurate and ready to use set of NLP tools. Its modularity and data centric architecture makes it flexible to pick and change or integrate new linguistic annotators. To add new modules the only requirement is to take KAF/NAF as standard input and, if the process is followed by another process in the

[16]http://uima.apache.org/

[17]http://nlp.stanford.edu/software/corenlp.shtml

[18]http://www.opener-project.org

[19]http://www.newsreader-project.eu

System	MUC	B^3	$CEAF_m$	BLANC	CONLL F1
Stanford	60.3	70.9	46.9	76.0	59.3
IXA Coref	55.1	68.5	45.6	71.5	56.4

Table 1: Evaluation of the Multi Sieve Pass on CoNLL 2011, *dev-auto* part.

chain, then to read KAF/NAF via standard output is also required.

Currently we offer linguistic annotation for English and Spanish, but more languages are being integrated. Furthermore, other annotations such as Semantic Role Labelling and Named Entity Disambiguation are being included in the pipeline following the same principles.

Additionally, current integrated modules are being improved: both on the quality and variety of the probabilistic models, and on specific issues such as lemmatization, and treatment of time expressions. Finally, we are adding server-mode execution into the pipeline to provide even faster processing. IXA pipeline is publicly available under Apache 2.0 license: *http://adimen.si.ehu.es/web/ixa-pipes*.

Acknowledgements

This work has been supported by the OpeNER FP7 project under Grant No. 296451, the FP7 NewsReader project, Grant No. 316404, and by the SKATER Spanish MICINN project No TIN2012-38584-C06-01. The work of Josu Bermudez on coreference resolution is supported by a PhD Grant of the University of Deusto (http://www.deusto.es).

6. References

Wauter Bosma, Piek Vossen, Aitor Soroa, German Rigau, Maurizio Tesconi, Andrea Marchetti, Monica Monachini, and Carlo Aliprandi. 2009. Kaf: a generic semantic annotation format. In *Proceedings of the GL2009 Workshop on Semantic Annotation*.

X. Carreras, L. Marquez, and L. Padro. 2002. Named entity extraction using AdaBoost. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–4.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 132–139. Association for Computational Linguistics.

Stephen Clark and James Curran. 2003. Language Independent NER using a Maximum Entropy Tagger. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-03)*, pages 164–167, Edmonton, Canada.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637.

Brooke Cowan and Michael Collins. 2005. Morphology and reranking for the statistical parsing of spanish. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 795–802. Association for Computational Linguistics.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. Gate: an architecture for development of robust hlt applications. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 168–175. Association for Computational Linguistics.

C. Fellbaum and G. Miller, editors. 1998. *Wordnet: An Electronic Lexical Database*. MIT Press, Cambridge (MA).

J. R. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370.

Jesús Giménez and Lluis Marquez. 2004. Svmtool: A general pos tagger generator based on support vector machines. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation*. Citeseer.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, pages 1–54, January.

D. Nadeau and S. Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):326.

J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. Curran. 2012. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, (In press).

Lluís Padró and Evgeny Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey, May. ELRA.

K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceed-*

ings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 492–501.

L. Ratinov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, page 147155.

Adwait Ratnaparkhi et al. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on empirical methods in natural language processing*, volume 1, pages 133–142.

Adwait Ratnaparkhi. 1998. *Maximum entropy models for natural language ambiguity resolution*. Ph.D. thesis, University of Pennsylvania.

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine learning*, 34(1-3):151–175.

Mariona Taulé, Maria Antònia Martí, and Marta Recasens. 2008. Ancora: Multilevel annotated corpora for catalan and spanish. In *LREC*.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of HLT-NAACL*, pages 252–259.

R. Weischedel, S. Pradhan, L. Ramshaw, J. Kaufman, M. Franchini, M. El-Bachouti, N. Xue, M. Palmer, M. Marcus, and A. Taylor. 2010. OntoNotes release 4.0. Technical report, Tech. rept. BBN Technologies.

Tong Zhang and David Johnson. 2003. A robust risk minimization based named entity recognition system. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 204–207. Association for Computational Linguistics.