Task-Driven Linguistic Analysis based on an Underspecified Features Representation

Stasinos Konstantopoulos,⋆ Valia Kordoni,† Nicola Cancedda,‡ Vangelis Karkaletsis,⋆ Dietrich Klakow,§ Jean-Michel Renders‡

⋆ Institute of Informatics and Telecommunications
NCSR 'Demokritos', Athens, Greece
{konstant,vangelis}@iit.demokritos.gr
† Department of Computational Linguistics and Phonetics,
Saarland University, Saarbrücken, Germany
and
Language Technology Lab, DFKI, Saarbrücken, Germany
kordoni@coli.uni-saarland.de
‡ Xerox Reseach Centre Europe, Grenoble, France
{Nicola.Cancedda,Jean-Michel.Renders}@xrce.xerox.com
§ Spoken Language Systems,
Saarland University, Saarbrücken, Germany
dietrich.klakow@lsv.uni-saarland.de

Abstract

In this paper we explore a *task-driven* approach to interfacing NLP components, where language processing is guided by the end-task that each application requires. The core idea is to generalize feature values into feature value *distributions*, representing under-specified feature values, and to fit linguistic pipelines with a back-channel of specification requests through which subsequent components can declare to preceding ones the importance of narrowing the value distribution of particular features that are critical for the current task.

Keywords: classification features, NLP component interfacing, task-driven language processing

1. Motivation

Although for many years the statistical revolution in computational linguistics overshadowed wide-coverage full-depth symbolic parsing systems, in recent years it has become clear that to obtain the kind of fine-grained syntactic and semantic analyses required for many applications, a judicious combination of deep analysis with statistically-trained models is needed.

In this position paper we explore the possibility of designing a new way of interfacing statistical and symbolic models, one that alleviates the need to either make performance concessions or face combinatorial explosion. The core idea is to pursue a *task-driven* approach where language processing is guided by the end-task that each application requires; and to achieve this by developing a novel architecture and interfacing between existing methods.

To better explain our research position, let us consider two extreme analysis pipelines: in one extreme, all analyses are carried over from each analysis stage to the next, as in the whiteboard architecture (Boitet and Seligman, 1994), resulting in a combinatorially-exploding task that can only be attained for very short texts or for controlled and almost unambiguous language usage. Alternatively, each stage needs to commit to the n-best analyses, as is the case in the DeepThought architecture (Callmeier et al., 2004), drastically pruning the search tree but also (especially in the earlier stages) making uninformed and arbitrary decisions which cannot be revised as more information becomes available. Many approaches estimate the confidence that an analysis is correct, or use some other method of ranking or filtering analyses based on syntactic correctness or semantic plausibility, but all are only as informed as previous analysis stages allow them to be.

Other approaches look beyond the sentence as the atomic unit of text under analysis, improving results by providing a context within which language is to be analysed. Discourse theories and contextual parsing methodologies help make more informed choices in all stages of the analysis, from choosing the most likely lexical lemma for each word, to choosing the most likely document class at the end of the pipeline. But the side-effect is that the unit of analysis has become longer, so that more aggressive pruning has become necessary: keeping many alternatives open at the discourse level is even more inefficient that keeping alternatives open at the sentence level and committing to a single context has even more serious repercussion for accuracy if the decision turns out to be wrong. In other words, these approaches are only as informed as the text analysed prior to the current analysis focus allows them to be.

Probabilistic settings partly alleviate the problem of making uninformed decisions by casting joint models of syntactic analysis and semantic interpretation (McCallum, 2009; Padó et al., 2009). Although these approaches explore new trade-offs between accuracy and efficiency, they do not avoid having to make accuracy concessions for the sake of efficiency.

In the remainder of this paper, we first present our research position on how to better organize linguistic pipelines and discuss the extensions needed in order to implement it (Sec-

tion 2.). We, then, proceed by surveying current language processing methodologies from the perspective of their compatibility with these extensions (Section 3.) and conclude by presenting our research programme (Section 4.).

2. Approach

In pursuing the goal stated above, we see two necessary steps: generalizing feature values into *feature value distributions* and providing linguistic pipelines with a back-channel of *specification requests* through which components can declare the importance of 'narrowing' the value distribution of particular features.

2.1. A backwards requests channel

For the purposes of demonstrating our point, let us assume a semantic classification task, where a linguistic analysis pipeline extracts sentence-level features from text, an interpretation module combines those into a semantic representation of the textual content, and a classifier uses these latter features to categorize the document.

We propose that instead of a purely forward-advancing pipeline, possibly back-tracking to get out of dead-end decisions, it is advantageous to have the classifier drive the whole process by deciding which features would mostly impact each classification decision; as opposed to having the classifier pick what it needs from what could be extracted. In other words, we envisage an architecture implementing (Figure 1):

- A backwards *requests channel* of information from the 'end-consumer' of features to the feature extraction modules, informing the latter on the features mostly needed to improve the former's confidence regarding the end-result. This allows extraction to focus computational resources on these mission-critical features.
- A forwards results channel of *underspecified features*, allowing feature extraction to defer committing to specific feature values and at the same time avoid branching off into a combinatorial search space.

After an initial analysis yields a largely underspecified feature vector, a request is generated with a ranked (or simply ordered) list of features that need to be more accurately specified. The ranking reflects an estimation of how critical it is, for the current classification task and given what features values are available right now, that a particular feature value is more tightly specified.

Upon receiving such feature specification requests, interpretation maps these to feature specification requests for the linguistic analysis component, specifying particular parse trees or fragments that need to be disambiguated. It is important to note that the interpretation module acts as a mediator or translator; that is to say, the interpretation module does *not* set requirements based on its own notion of how to improve the quality of the interpretation. It, rather, uses its knowledge of the dependencies between its input and output features in order to 'translate' the classifier's request into terms that are pertinent to linguistic analysis.

Upon receiving this request, linguistic analysis initiates a new iteration focusing the features where a tighter approximation has been requested.

The major advantage is that the system neither forces modules to commit to uninformed choices nor requires extensive combinatorial searches. The individual modules retain complete models of alternative analyses under consideration, but these are not propagated to the next analysis stage; they are rather iteratively refined until the classifier is able to, confidently enough, provide results.

2.2. Representing underspecified features

One of the critical aspects of this methodology is the efficient interfacing of the analysis modules, both at the conceptual level of representing intermediate analysis results as well as the concrete level of an efficient implementation. More specifically, although the overall methodology remains agnostic of the particularities of the representations internally used by the various components, it requires that components are able to export analysis results as *underspecified features*, numeric as well as symbolic. The representation for such underspecified features should be geared towards relaying *meta-information* about features, such as the range of possible values and the distribution of a metric of confidence along this range.

The key is that alternative analyses are not exhaustively searched, or even shared with other components. Instead, the current analysis at each iteration is an approximation, represented as underspecified features. Components will operate upon underspecified features to offer a new confidence distribution as a result. In our classification example, for instance, interpretation can be based on many-valued semantic inference.

The final component in the pipeline, the features 'end-user', should, besides being able to operate upon underspecified features, also be able to choose which features cause the widest variation in the end-result and should be more tightly approximated. In our example, the classifier evaluates the value distributions of the features based on the perturbation they cause in the class the document falls under. In this manner, a very wide distribution in one feature might be acceptable if it mostly leads to the same document class (given the value distributions of the other features), whereas a relatively tight distribution might need to be refined if the document class is sensitive to even small variations in this feature's values.

This initiates a new iteration, where each component exploits knowledge of the inter-dependencies between features that its own methodology introduces in order to map feedback coming from the component after it in the pipeline into feedback for the component placed before it. That is, it translates a request referring to the distribution over its output features into a request referring to the distribution over its input features.

In order to support the flexibility necessary to become the basis of a variety of language technology systems, and in order to allow for building pipelines that also include components that do not conform to this representation, especially in the early phases of our research programme, care must also be taken to design for *extendability*. More specifically, the representation should provide a mechanism of extensions specific to certain component types (e.g., syntactic analysers) or even specific component implementa-

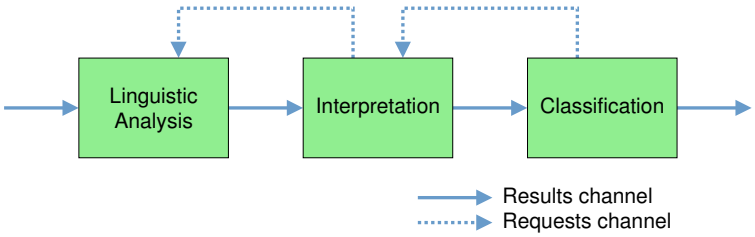

Figure 1: Outline of the architecture.

tions. This will allow for more detailed annotations where relevant and where all components involved are knowledgeable of these extensions. Naturally, such extended annotations should be in addition to (and not instead of) and should be consistent with the basic values distribution.

3. Background

As stated in the introductory section, it is desired that our research programme is about architecture and interfacing and does not require any substantial extensions to current NLP methods. In this section we briefly argue about how current methods can support the advocated re-organization of the dependencies between components.

3.1. Classification

Classification tasks have seen various alternative formalizations in the literature: as binary decisions, as a ranking task among multiple events, as a regression task aimed at providing a quantitative score, and more.

As the final consumer of features, the methods that are are consistent with our methodology must offer themselves to the estimation of (a) the accuracy that can be achieved by the current feature specificity, and (b) the relevance of the different features to each particular decision. In this manner, the classifier can thus decide that, for example, although the values of many features are very unspecific, enough specific features are present to make the rest of the feature values gratuitous. If not, it estimates which features are dominant in the decision currently at hand (taking also into account feature values which are already specific) and requests that they are more tightly approximated.

Conceptually, this is akin to well-studied decision problems, such as deciding what test to perform on patients in order to establish what condition they are suffering from. This makes frameworks of statistical decision theory (Berger, 1985) and Markov decision processes (Puterman, 2005) most amenable, using the entropy of the distribution over class membership as the utility function of improving the approximation of a given feature.

Recent work on *confidence-weighted linear classifiers* is also relevant, as feature meta-information is an inherent component of the approach. Confidence-weighted classification was introduced (Dredze et al., 2008; Dredze et al., 2010) in order to handle rare features well, as memoryless classifiers do not distinguish between reliable features whose weight have been tuned in the course of many updates and feature weights estimated by only a few updates. Finally, generalized linear models (Nelder and Wedderburn, 1972), and in particular their Bayesian treatment by Gelman et al. (1995), provide a theoretically well-founded basis for confidence weighting of the model vectors.

3.2. Document Interpretation

Document understanding is often approached as a semantic inference task, fusing the various pieces of information extracted by sentence-level NLP and combining it with background knowledge in order to yield an abstract interpretation of the whole document.

The key properties that this inference process needs in order to be compatible with our framework is that it can:

- operate upon the underspecified features representation to yield underspecified structured information from syntactically and semantically analysed text; and
- operate in the reverse direction to map classifier requests into requests for the NLP module.

Although text analysis has been approached from both purely rule-based and purely statistical angles, approaches that combine knowledge-based techniques with machine learning have been gaining ground ever since such hybrid systems were ranked at the top of the Named Entity task at MUC-7 (Mikheev et al., 1998). Such systems combine feature-extraction rules with some notion of numerical valuation (probabilistic or other) in order to convey and treat the confidence in the accuracy of each individual feature extracted; and to take this valuation into account when raking alternative interpretations of the text (complete feature vectors).

The emergence of the Semantic Web has brought *ontologies* and related Semantic Web formalisms in the pole position for encoding knowledge in knowledge-based systems, either targeting uni-modal text analysis or multi-modal document analysis where text is one of the modalities (Nédellec and Nazarenko, 2005; McDowell and Cafarella, 2006; Buitelaar et al., 2008; Fragkou et al., 2008). Interestingly, and despite the fact that probabilistic and fuzzy extensions of ontological inference are already in place, hybrid systems akin to those in the last MUC competitions are not reported in the literature: ontology-based extraction typically amounts to using ontologies as a terminology that facilitates term extraction, preferring other modes of inference to identify the semantic relations between these terms such as crisp logical inference (Aitken, 2002; Espinosa Peraldi et al., 2008), support vector machines (Zhou et al., 2005; Hong, 2005), or conditional random fields (Choi et al., 2005).

Symmetrically, the logical inference community sees fuzzy ontologies as an opportunity to encode the semantics of

vague language (Lukasiewicz and Straccia, 2007), and have not explored their potential for encoding feature extraction rules.

We argue, however, that this potential is considerable and worth exploring: uncertainty inference systems combine logical inference with numerical constraint satisfaction to draw conclusions over partially known numerical data (Konstantopoulos and Apostolikas, 2007; Bobillo and Straccia, 2008), which fits perfectly both exploiting ontological terminologies and other semantic background knowledge and operating over underspecified features. In the reverse direction, computational intelligence methods such as error back-propagation have been used to distribute user feedback among the unimodal sub-components of multimodal information fusion systems (Apostolikas and Konstantopoulos, 2007). Such methods fit perfectly the role of deciding which features should be more tightly approximated in order to satisfy incoming requests.

3.3. Text analysis

The core concepts required to support our approach have already been instroduced in text analysis: that efficient but low-accuracy first-pass attempts at analysis can be followed up by more detailed analysis; and that commiting to a specific analysis be delayed until more information is available.

With respect ot the first, modern full-depth syntactic parsers often rely on a shallow parsing pre-processing step. Furthermore, the full-depth representation allows for spans of shallow-parsed text to be embedded in parse trees. Such spans are only specified by a flat label with any deeper feature structures absent.

Such shallow results are currently used as fallback to increase system robustness, but for our purposes can also be used as a first approximation. In such a situation, shallowly-extracted flat labels would map to largely underspecified feature structures. Upon receiving a request to specify a given feature, the corresponding span can be deeply parsed. In this manner, the parser can be directed, for example, towards specific anaphora instances that need to be resolved in order to extract critical features and away from resolving ambiguities that do not contribute to the classification.

Regarding the second point that a text analysis component can delay decisions, many modern parsers use *packed representations*, allowing to efficiently maintain alternative analyses and defer for later iterations decisions about which analysis to propagate to next stage. One such representation is the one used in the PET parser (Callmeier et al., 2004), implementing an unpacking algorithm that extracts the n-best analyses from a packed parse forest in time linear to the number of the *retrieved* parse trees, regardless of the size of the forest (Zhang et al., 2007; Zhang and Kordoni, 2010).

Similarly, Kim et al. (2010) discuss the same trade-off between combinatorial explosion and uninformedly commiting to solutions in the context of ambiguity resolution that is the focus of this paper. In order to address this, they propose an extension of the Stanford parser (Klein and Manning, 2003) with intensional representation of ambiguities and the dependencies among them. The representation maintains alternatives while avoiding enumerating them and is coupled with pruning and update algorithms that operate directly on the packed representation.

4. Conclusions

We explored the possibility of a *task-driven* approach to text understanding, where the process is guided by the application for which it is needed. We believe this to be an advantageous alternative to current NLP pipelines, especially for larger and more complex documents where the overhead that our approach introduces will be offset by the efficiency gained by not extracting gratuitous features. Furthermore, since some of the computations in the process are independently needed in some situations, e.g., maintaining feature meta-information for model adaptation, we believe that we can achieve substantial performance improvements in many real-world applications.

Besides describing the core idea, we have briefly surveyed current language processing, semantic inference, and document classification methods and have demonstrated that our approach can be implemented with minimal extensions to currently common approaches.

As a first step in our research programme, we will thus test our idea on the application discussed in this paper in order to empirically validate and measure the expected efficiency gain. This testbed can also be used to research further questions, such as:

- Identifying appropriate termination criteria and mechanisms for the iterative process.
- Developing criteria and mechanisms for deciding how far back feedback should be pushed through the pipeline. That is, deciding whether feedback is to be most appropriately handled by this component or by one of the components before it.

Further steps would include identifying other applications besides document classification where it can be applied. *Speech processing,* in particular, would be an immediate objective, since it introduces another major factor of ambiguity before parsing. In particular, we expect our method to greatly increase the size of the *active vocabulary* that speech recognizers can use without leading to too many false positive recognitions to be usable.

Finally, working out the interactions between our approach and discourse-level analysis or dialogue management would be an interesting challenge.

5. References

James Stuart Aitken. 2002. Learning information extraction rules: An Inductive Logic Programming approach. In *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI '02)*.

Georgios Apostolikas and Stasinos Konstantopoulos. 2007. Error back-propagation in multi-valued logic systems. In *Proceedings of the 7th International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), Sivakasi, Tamil Nadu, India, 13–15 December*, volume IV, pages 207–213. IEEE CS Press.

James O. Berger. 1985. *Statistical Decision Theory and Bayesian Analysis*. Springer.

Fernando Bobillo and Umberto Straccia. 2008. fuzzyDL: an expressive fuzzy Description Logic reasoner. In *Proceedings of the 2008 International Conference on Fuzzy Systems (FUZZ-08)*.

Christian Boitet and Mark Seligman. 1994. The whiteboard architecture: A way to integrate heterogeneous components of NLP systems. In *Proceedings of the 15th Conference on Computational Linguistics (COLING-94)*.

P. Buitelaar, P. Cimiano, A. Frank, M. Hartung, and S. Racioppa. 2008. Ontology-based information extraction and integration from heterogeneous data sources. *International Journal of Human-Computer Studies*, 66(11):759–788.

Ulrich Callmeier, Andreas Eisele, Ulrich Schäfer, and Melanie Siegel. 2004. The DeepThought core architecture framework. In *Proc. 4th Intl Conf. on Language Resources and Evaluation (LREC-04). Lisbon*.

Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP 2005), Vancouver, BC, Canada, 6–8 October*.

Mark Dredze, Koby Crammer, and Fendando Pereira. 2008. Confidence-weighted linear classification. In *Proceedings of the International Conference on Machine Learning (ICML-08)*.

Mark Dredze, Alex Kulesza, and Koby Crammer. 2010. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning Journal*, 79(1–2):123–149, May.

Irma Sofia Espinosa Peraldi, Atila Kaya, and Ralf Moller. 2008. Formalizing multimedia interpretation based on abduction over Description Logic ontologies. In *Proceedings of the 3rd International Conference on Semantic and Digital Media Technologies, Koblenz, Germany, 3–5 December*.

Pavlina Fragkou, Georgios Petasis, Aris Theodorakos, Vangelis Karkaletsis, and Constantine D. Spyropoulos. 2008. BOEMIE ontology-based text annotation tool. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008), Marrakech, Morocco, May 2008*.

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. 1995. *Bayesian Data Analysis*. Chapman and Hall.

Gumwon Hong. 2005. Relation extraction using support vector machine. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05), Jeju Island, Korea, 11–13 October*, volume 3651 of *Lecture Notes in Computer Science*.

Doo Soon Kim, Ken Barker, and Bruce Porter. 2010. Improving the quality of text understanding by delaying ambiguity resolution. In *Proceeding of the 23rd International Conference on Computational Linguistics (COLING '10), Beijing, China, 23–27 August*.

Dan Klein and Chris Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03), Sapporo, Japan, 7-12 July*, pages 423–430.

Stasinos Konstantopoulos and Georgios Apostolikas. 2007. Fuzzy-DL reasoning over unknown fuzzy degrees. In *Proc. Intl IFIP Workshop of Semantic Web and Web Semantics (IFIP-SWWS 07), Vilamoura, Portugal*, volume 4806 of *Lecture Notes in Computer Science*. Springer.

Thomas Lukasiewicz and Umberto Straccia. 2007. Description logic programs under probabilistic uncertainty and fuzzy vagueness. In *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU-07), Hammamet, Tunisia, 31 Oct–2 November*.

Andrew McCallum. 2009. Joint inference for natural language processing. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL-09)*.

L. K. McDowell and M. Cafarella. 2006. Ontology-driven information extraction with OntoSyphon. In *Proceedings of the 5th International Semantic Web Conference (ISWC-06), Athens, GA, USA, 5–9 November*, volume 4273 of *Lecture Notes in Computer Science*. Springer.

Andrei Mikheev, Claire Grover, and Marc Moens. 1998. Description of the LTG system used for MUC-7. In *Proceedings of 7th Message Understanding Conference (MUC-7)*.

Claire Nédellec and Adeline Nazarenko. 2005. Ontology and information extraction: A necessary symbiosis. *Ontology Learning from Text: Methods, Evaluation and Applications*, 123.

John Nelder and Robert Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society, Series A*, 135(3):370–384.

Ulrike Padó, Matthew W. Crocker, and Frank Keller. 2009. A probabilistic model of semantic plausibility in sentence processing. *Cognitive Science*, 33(5):794–838.

Martin L. Puterman. 2005. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley.

Yi Zhang and Valia Kordoni. 2010. Discriminant ranking for efficient treebanking. In *Proc. of the 23rd Intl Conf. on Computational Linguistics (COLING 2010). Beijing, China*.

Yi Zhang, Stephan Oepen, and John Carroll. 2007. Efficiency in unification-based n-best parsing. In *Proceedings of the 10th Intl Conf. on Parsing Technologies (IWPT-07). Prague, Czech Rep*.

GuoDong Zhou, Su Jian, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting the the Association for Computational Linguistics (ACL-05), Ann Arbor, Michigan, USA, 25–30 June*.