Empirical Learning of Natural Language Processing Tasks

Walter Daelemans[1], Antal van den Bosch[2], Ton Weijters[2]

[1] Computational Linguistics, Tilburg University, The Netherlands
[2] Dept. of Computer Science / MATRIKS, Universiteit Maastricht, The Netherlands

Abstract. Language learning has thus far not been a hot application for machine-learning (ML) research. This limited attention for work on empirical learning of language knowledge and behaviour from text and speech data seems unjustified. After all, it is becoming apparent that empirical learning of Natural Language Processing (NLP) can alleviate NLP's all-time main problem, viz. the knowledge acquisition bottleneck: empirical ML methods such as rule induction, top down induction of decision trees, lazy learning, inductive logic programming, and some types of neural network learning, seem to be excellently suited to automatically induce exactly that knowledge that is hard to gather by hand. In this paper we address the question why NLP is an interesting application for empirical ML, and provide a brief overview of current work in this area.

1 Empirical Learning of Natural Language

Looking at the ML literature of the last decade, it is clear that language learning has not been an important application area of ML techniques. Especially the absence of more research on the *empirical learning of language knowledge and behaviour from text and speech data* is strange. After all, a main problem of the AI discipline of Natural Language Processing (NLP) is the *knowledge-acquisition bottleneck*: for each new language, domain, theoretical framework, and application, linguistic knowledge bases (lexicons, rule sets, grammars) have to be built basically from scratch.

In our opinion, there are at least three reasons why ML researchers should become more interested in NLP as an application area:

- **Complexity of tasks.** Data sets describing language problems exhibit a complex interaction of regularities, sub-regularities, pockets of exceptions, idiosyncratic exceptions, and noise. As such, they are a perfect model for a large class of other poorly-understood real-world problems (e.g. medical diagnosis) for which it is less easy to find large amounts of data. A better understanding of which algorithms work best for this class of problems will transfer to many other problem classes.
- **Real-world applications.** The market pull for applications in NLP (especially text analysis and machine translation) is enormous, but has not been matched by current language technology. ML techniques may help in realising the enormous market potential for NLP applications.

- **Availability of large datasets.** Datasets of NLP tasks containing tens or hundreds of thousands of examples are readily available. Traditional benchmark datasets usually contain far less examples. Experimenting with linguistic datasets will force algorithm designers to work on the issue of scaling abilities.

Examples of linguistic data available on electronic media exist across the board. Corpora that have been made available for research purposes and which can be or already have been used for empirical learning of NLP tasks include the following[3]:

- **CELEX** is a lexical data base containing word lists of English, German, and Dutch: for each word, detailed phonological and morphological information is provided. For each language, information on a hundred thousand words or more is available;
- **Penn Treebank II** is a data base of parsed and tagged sentences, stemming mainly from the Wallstreet Journal, containing about a million words;
- **WordNet** is a lexical data base for English, in which relations between words are implemented as links in a semantic network.

Our optimism about the marriage of empirical learning and NLP is based on our claim that NLP tasks fit the classification paradigm of supervised ML very well. Empirical learning (inductive learning from examples) is fundamentally a *classification* paradigm. Given a description of an object in terms of a propositional or first-order language, a category label is produced. This category should normally be taken from a finite inventory of possibilities, known beforehand. It is our claim that *all* linguistic tasks can be redefined this way and can thus be taken on in a ML context. All linguistic problems can be described as a mapping of one of two types of classification (Daelemans, 1995):

- **Disambiguation.** Given a set of possible categories and a relevant context in terms of attribute values, determine the correct category for this context. An example from text-to-speech conversion: given a letter in its context (a word), determine its pronunciation. An example from parsing: given a word in a sentence, determine the syntactic role of the word.
- **Segmentation.** Given a target and a context, determine whether a boundary is associated with this target, and if so which one. An example from word processing: given a position in a word, determine whether the word can be hyphenated there. An example from parsing: given two words in a sentence, determine whether a syntactic constituent boundary lies between the words.

[3] The three corpora can be reached at URLs http://www.kun.nl/celex/ (CELEX); ftp://ftp.cis.upenn.edu/pub/treebank/public_html/home.html
(Penn Treebank); and http://www.cogsci.princeton.edu/~wn/ (WordNet). Cf. http://www.cs.unimaas.nl/signll/signll-www.html for more links to home pages of corpora.

To redefine linguistic tasks as classification tasks appears straightforward for tasks such as text-to-speech conversion and hyphenation (i.e., tasks in the *morpho-phonological* domain), but may appear less so for complex NLP tasks such as word-sense disambiguation or parsing (i.e., tasks in the *syntactic-semantic* domain). Such complex tasks should not be redefined as one-pass classification tasks (e.g., given a sentence of written words, determine whether it is grammatical), but they can be defined as *cascades* of disambiguation and segmentation tasks. For example, parsing can be decomposed into deciding on the morpho-syntactic role of words (disambiguation), finding constituent boundaries (segmentation), resolving attachment ambiguities, determining the label of constituents, and determining grammaticality of sequences of constituents (all three disambiguation). Besides studying the learnability of identified linguistic tasks (which is what most current work is aimed at), an additional research goal of empirical learning of NLP tasks is therefore to search and test appropriate decompositions of complex tasks into tasks which are more easily learnable.

In the remainder of this paper, we provide an overview of current research on the empirical learning of NLP tasks. While the amount of work in some domains is limited, the results are often impressive. We structure the overview in four sections. This structure reflects what we view as an important dimension in empirical learning of NLP tasks, of which *lazy learning* on the one hand, and *greedy learning* on the other hand are the extremes. The essential difference between the two extremes is that in lazy learning, information encountered in training is not abstracted, whereas in greedy learning information is abstracted by restructing and removing redundant or unimportant information. Applications of lazy-learning algorithms are given in Section 2. The next three sections describe three approaches to greedy learning, viz. decision-tree learning and rule induction (section 3, artificial neural networks (section 4, and inductive logic programming (section 5).

2 Lazy Learning

The lazy learning learning paradigm is founded on the hypothesis that performance in cognitive tasks (in our case language processing) is based on reasoning on the basis of analogy of new situations to *stored representations of earlier experiences*, rather than on the application of *mental rules* abstracted from earlier experiences (as in rule induction and rule-based processing). The concept has appeared in different AI disciplines (from computer vision to robotics) several times, using alternative terms such as similarity-based, example-based, exemplar-based, analogical, case-based, nearest-neighbour, instance-based, and memory-based (Stanfill and Waltz, 1986; Kolodner, 1993; Aha *et al.*, 1991). Learning is 'lazy' as it involves adding training examples (feature-value vectors with associated categories) to memory without abstraction or restructuring. During classification, a previously unseen test example is presented to the system. Its similarity to all examples in memory is computed using a *similarity metric*, and the category of the most similar example(s) is used as a basis for predicting the category of the test example.

From the early nineties onwards, lazy-learning approaches to NLP tasks have been explored intensively by the partners of the ATILA project (University of Tilburg, Antwerp University, Universiteit Maastricht). Daelemans (1995) provides an overview of early work of this group on phonological and morphological tasks (grapheme-to-phoneme conversion, syllabification, hyphenation, morphological synthesis, word stress assignment). More recently, the approach has been applied to part-of-speech tagging (morphosyntactic disambiguation), morphological analysis, and the resolution of structural ambiguity (PP-attachment) (Daelemans *et al.*, 1996; Van den Bosch *et al.*, 1996). Cardie (1994, 1996) suggests a lazy-learning approach for both (morpho)syntactic and semantic disambiguation and shows excellent results compared to alternative approaches. Ng and Lee (1996) report results superior to previous statistical methods when applying a lazy learning method to word sense disambiguation. The exemplar-based reasoning aspects of lazy learning are also prominent in the large literature on example-based machine translation (see Jones, 1996, for an overview).

3 Decision-Tree Learning and Rule Induction

The *decision-tree learning* paradigm is based on the assumption that similarities between examples can be used to automatically extract decision-trees and categories with explanatory and generalisation power. In this paradigm, learning is *greedy*, and abstraction occurs at learning time. Decision-tree learning is a well-developed field within AI, see e.g. Quinlan (1993) for a synthesis of major research findings. The goal of *rule induction* (e.g., C4.5rules, Quinlan, 1993; CN2, Clark and Boswell, 1991) is, more than it is with decision-tree learning, to induce limited sets of interpretable rules from examples or decision trees.

Work on parsing (including tagging) of text with decision trees was pioneered at IBM (Black *et al.*, 1992, Magerman, 1995). SPATTER (Magerman, 1995) starts from the premise that a parse tree can be viewed as the result of a series of classification problems. The most probable sequence of decisions for a sentence, given a training corpus, is its most probable analysis. Schmid (1994) describes TREETAGGER, in which transition probabilities between tags in a tag sequence are estimated using a decision tree induced from a set of n-grams occurring in the Penn treebank corpus. The features are the tags of the words preceding the word to be tagged. Schmid reports robustness relative to training set size: TREETAGGER 'degrades gracefully' with smaller training set sizes.

An example application of rule induction to the semantic domain is the classification of dialogue acts (Andernach, 1996). In this work, rule induction is employed to automatically generate and test theories on what are useful cues in texts for classifying them as dialogue acts. The output of rule induction offers interesting alternative insights to what existing theories consider relevant (Andernach, 1996). The use of rule induction to find heuristics for disambiguating between discourse use and sentential use of cue phrases in text was investigated by Litman (1996).

In the morpho-phonological domain the decision-tree learning algorithm IG-Tree (Daelemans *et al.*, to appear) has been applied successfully to grapheme-phoneme conversion (Van den Bosch and Daelemans, 1993) and morphological analysis (Van den Bosch *et al.*, 1996). An example application of rule induction to a morpho-phonological task is the application of C4.5rules to Dutch diminutive formation (Daelemans *et al.*, 1996).

4 Artificial Neural Networks

During the last decade, the study of connectionist models or *Artificial Neural Networks* (ANNs), has also led to applications in the NLP domain. The type of ANN learning most commonly used for NLP tasks, viz. supervised learning of classification tasks, contrasts with symbolic approaches with respect to its non-symbolic knowledge representation. The functionality of trained ANNs nonetheless displays the same interesting properties as that of lazy learning and decision-tree learning: an ANN can represent abstractions as well as store specific input-output mappings. A commonly-used learning algorithm for supervised learning of classification tasks in ANNs is back-propagation (BP) (Rumelhart, Hinton, and Williams, 1986).

In the current development of applying ANNs in NLP, one finds a stress on the issue of *representation* in syntactic and semantic applications, and on *generalisation* in morpho-phonological applications. Successful applications to syntax and semantics include modelling state machines discriminating between grammatical and ungrammatical sentences (e.g., Lawrence, Fong, and Giles, 1995). For excellent overviews of ANN applications to syntax and semantics, the reader is referred to Reilly and Sharkey (1992), and Wermter, Riloff, and Scheler (1996).

In the morpho-phonological domain, successes are claimed for ANNs as good generalisation models for classification tasks, e.g., grapheme-phoneme conversion (Dietterich *et al.*, 1995; Wolters, 1995). However, work by Weijters (1991), Van den Bosch and Daelemans (1993), and Van den Bosch (forthcoming), consistently shows a significantly lower performance on a range of morpho-phonological subtasks by BP as compared to decision-tree learning and lazy learning. Apparently, the amount of abstraction in a BP-trained ANN is accounting for a similar harmful effect on generalisation performance witnessed in decision-tree learning as opposed to lazy learning.

5 Inductive Logic Programming

Inductive Logic Programming (ILP) is one of the newest subfields in AI. For a general introduction to ILP, see Lavrac and Dzeroski (1994), or the contribution of Muggleton and De Raedt (1994) in an anniversary issue of the Journal of Logic Programming. ILP algorithms induce first-order hypotheses from examples. By using first-order logic as representation language, ILP can successfully learn problems for which feature-value-based algorithms fail.

First-order logic plays a crucial role in ILP. The general aim is to induce a hypothesis such that the classification of each learning example is entailed by the combination of background knowledge, the induced hypothesis, and the example. First-order (clausal) logic is used for the description of the background knowledge, the learning examples, and the hypothesis.

Despite the limited number of applications in which the relatively novel method of ILP is used for NLP tasks, the results are impressive: the rich representation language and the use of background knowledge in ILP enables the learning of complex NLP tasks such as (semantic) parsing (Zelle and Mooney, 1994; Muggleton *et al.* 1996), and tagging (Cussens, 1996). Dehaspe *et al.* (1996) uses ILP for small-scale linguistic tasks: grammaticality checking and Dutch diminutive forming.

6 Conclusion

Some general trends become clear when analysing the results of these studies. First, the most striking result is that the accuracy of induced systems is always comparable and often better than state-of-the-art hand-crafted systems, at a fraction of the development effort and time. This proves the point that ML techniques may help considerably in solving knowledge acquisition bottlenecks in NLP.

Second, it depends on the goal of the system whether lazy learning or greedy learning algorithms at an advantage. If the goal is optimal accuracy, lazy learning is preferable (Daelemans, 1996). We find that simple lazy-learning algorithms, extended with feature weighting and probabilistic decision rules, consistently obtain the best generalisation accuracy on a large collection of linguistic tasks (e.g., within the morpho-phonological domain, Van den Bosch, forthcoming). A possible explanation for this is the structure of NLP tasks discussed earlier: apart from a number of clear generalisations, a lot of subregularities and exceptions exist in the data. Exceptions tend to come in 'families'; it is therefore advantageous to *keep exceptions* (some family members may turn up during testing) rather than abstracting away from them: *being there is preferable to being probable.* If the goal of learning is creating inspectable, understandable generalisations about the data, however, the greedy-learning algorithms are obviously at an advantage: greedy learning techniques, such as ILP and Rule Induction, induce structures which may add to the understanding of the domain, and indeed sometimes generate new linguistic descriptions of the domain.

Third, the learning techniques described are well-suited for *integrating* different information sources (e.g. syntactic and semantic features may be combined in a single feature vector). Especially in lazy learning, feature-weighting techniques in the similarity metric achieve an optimal fusion and integration of these information sources in many applications.

The study of the usefulness of empirical ML for NLP has only just begun, but the results already achieved certainly warrant further systematic investigation.

References

Aha, D., Kibler, D., and Albert, M. (1991). Instance-based learning algorithms. *Machine Learning*, 7:37–66.

Andernach, T. (1996) A machine learning approach to the classification of dialogue utterances. In K. Oflazer and H. Somers (Eds.), *Proceedings of the Second International Conference on New Methods in Language Processing*, NeMLaP, pp. 98–109.

Black, E., Jelinek, F., Lafferty, J, Mercer, R., and Roukos, S. (1992). Decision tree models applied to the labeling of text with parts-of-speech Darpa Workshop on Speech and Natural Language.

Cardie, C. (1994). *Domain-Specific Knowledge Acquisition for Conceptual Sentence Analysis.* Ph.D. Thesis, University of Massachusetts, Amherst, MA.

Cardie, C. (1996). Embedded Machine Learning Systems for Natural Language Processing: A General Framework. In S. Wermter, E. Riloff, and G. Scheler, (Eds.), *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, pp. 315–328. Berlin: Springer-Verlag.

Clark, P., and Boswell, R. (1991). Rule induction with CN2: Some recent improvements. In *Machine Learning: Proceedings of the Fifth European Conference*, pp. 151-163.

Cussens, J. (1996). Part-of-speech disambiguation using ILP. Technical report PRG-TR-25-96, Oxford University Computing Laboratory.

Daelemans, W. (1995). Memory-based lexical acquisition and processing. In P. Steffens (Ed.), *Machine Translation and the Lexicon*, pp. 85–98. Berlin: Springer-Verlag.

Daelemans, W. (1996). Abstraction considered harmful: Lazy learning of language processing. In H. J. van den Herik and A. Weijters (Eds.), *Proceedings of the 6th Belgian-Dutch Conference on Machine Learning*, Maastricht, The Netherlands, pp. 3–12.

Daelemans, W., Berck, P, and Gillis, S. (1996). Unsupervised discovery of phonological categories through supervised learning of morphological rules. In *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, pp. 95-100.

Daelemans, W., Van den Bosch, A., and Weijters, A. (to appear). IGTree: using trees for classification in lazy learning algorithms. *Artificial Intelligence Review*, special issue on Lazy Learning. To appear.

Dehaspe, L., Blockeel, H., and De Raedt, L. (1996). Induction, logic and natural language processing. In *Proceedings of the Joint* ELSNET/COMPULOG-NET/EAGLES *Workshop on Computational Logic for Natural Language Processing*, South Queensferry, Scotland.

Dietterich, T. G., Hild, H., and Bakiri, G. (1990). A comparison of ID3 and Backpropagation for English text-to-speech mapping. Technical Report 90–20–4, Oregon State University.

Jones, D. *Analogical Natural Language Processing.* London: UCL Press, 1996.

Kolodner, J. (1992). *Case-Based Reasoning.* San Mateo, CA: Morgan Kaufmann.

Lavrac, N. and Dzeroski, S. (1994). *Inductive Logic Programming.* Chichester, UK: Ellis Horwood.

Lawrence, S., Fong, S., and Giles, C. Lee (1991) Natural language grammatical inference: A comparison of recurrent neural networks and machine learning methods. In S. Wermter, E. Riloff, and G. Scheler (Eds.), *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, pp. 33–47. Berlin: Springer-Verlag.

Litman, D. J. (1996). Cue phrase classification using machine learning. *Journal of Artificial Intelligence Research*, 5:53-94, 1996.

Magerman, D. (1995). Statistical decision tree models for parsing. In *Proceedings of the Association for Computational Linguistics.*, 1995.

Muggleton, S., and De Raedt, L. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19,20:629-679.

Muggleton, S., Page, D., and Srinivasan, A. (1996). Learning to read by theory revision. Technical Report PRG-TR-26-96, Oxford University Computing Laboratory.

Ng, H. T. and H. B. Lee (1996). Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In *Proceedings of the annual meeting of the ACL*, ACL-96.

Quinlan, J. R. (1993). C4.5*: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Reilley, R. G. and Sharkey, N. E., Eds. (1992). *Connectionist Approaches to Natural Language Processing.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pp. 318–362. Cambridge, MA: The MIT Press.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of the International Conference on New Methods in Language Processing*, NeMLaP, Manchester, 44–49.

Van den Bosch, A., and Daelemans, W. (1993). Data-oriented methods for grapheme-to-phoneme conversion. In *Proceedings of the 6th Conference of the EACL*, pp. 45–53.

Van den Bosch, A., Daelemans, W., and Weijters, A. (1996). Morphological analysis as classification: an inductive-learning approach In K. Oflazer and H. Somers (Eds.), *Proceedings of the Second International Conference on New Methods in Language Processing*, NeMLaP, Ankara, Turkey, pp. 79–89.

Van den Bosch, A. (forthcoming). *Machines Learning to Pronounce Words: Empirical Learning and Modularisation of Morpho-Phonology.* PhD-Thesis, Universiteit Maastricht.

Weijters, A. (1991). A simple look-up procedure superior to NETtalk? In *Proceedings of the International Conference on Artificial Neural Networks*, Espoo, Finland.

Wermter, S., Riloff, E., and Scheler, G. (1996). *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing.* Berlin: Springer-Verlag.

Wettschereck, D., Aha, D. W. & Mohri, T. (1996). A review and comparative valuation of feature weighting methods for lazy learning algorithms. Technical Report AIC-95-012. Washington, DC: NRL Navy Center for Applied Research in AI.

Wolters, M. (1995). A dual-route approach to grapheme–to–phoneme conversion. In *Proceedings of the International Conference on Artificial Neural Networks.*

Zelle, J. M., and Mooney, R. J. (1994). Inducing deterministic Prolog parsers from treebanks: A machine learning approach. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, WA, pp. 748–753.