Resource Monitoring in Information Extraction

Jochen L. Leidner[1,2]

[1]) Linguit Ltd.
18/1 Bruntsfield Avenue
Edinburgh EH10 4EW, Scotland, UK.
⟨http://www.linguit.com⟩
leidner@linguit.com

[2]) University of Edinburgh
School of Informatics
2 Buccleuch Place, Edinburgh EH8 9LW,
Scotland, UK.
jochen.leidner@ed.ac.uk

ABSTRACT

It is often argued that in information extraction (IE), certain machine learning (ML) approaches save development time over others, or that certain ML methods (e.g. Active Learning) require less training data than others, thus saving development cost. However, such development cost claims are not normally backed up by controlled studies which show that such development cost savings actually occur. This situation in Language Engineering (LE) is contrasted with Software Engineering in general, where a lot of studies investigating system development cost have been carried out. We argue for the need of controlled studies that measure actual system development time in LE. To this end, we carry out an experiment in resource monitoring for an IE task: three named entity taggers for the same "surprise" domain are developed in parallel, using competing methods. Their human development time is accounted for using a logging facility. We report development cost results and present a breakdown of the development time for the three alternative methods. We are not aware of detailed previous parallel studies that detail how system development time in IE is spent.

Categories and Subject Descriptors

D.2.8 [**Software**]: Software Engineering—*Metrics: Process metrics*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing

Keywords

methodology; language engineering economics; NERC; cost metrics; machine learning; named entity tagging

1. INTRODUCTION

Computational Linguistics as a scientific discipline uses methods to obtain knowledge of human language processing that must be subjected to repeated falsification attempts based on criteria of descriptive and explanatory adequacy, among others, but which is not affected by practical concerns such as computational efficiency. Language Engineering, on the other hand, is a technological discipline. Information Extraction, and its most essential part, Named Entity Recognition and Classification (NERC), are part of this technological (as opposed to purely knowledge-seeking) realm, and like for any engineering discipline, they should thus abide by engineering principles. One principal desideratum of engineering work is that the engineer develops a product or service *to a specification*, which acts as a set of external constraints imposed on him or her from the outside [2]. Such constraints reflect limited resources in the physical environment in which engineering work takes place: time is precious (most projects are of short duration), the number of staff allocated to a project is limited (and increasing staff count decreases productivity), and most projects have rigid budgets. The goal of this study is to assess empirically the development cost of named entity taggers for the same task, using three alternative methods.

1.1 Related Work

In Information Extraction, data about system development efforts are not easily available. Riloff has shown, for instance, that dictionaries for a typical IE tasks can be extracted in as little as 5 person hours using a bootstrapping approach [6]; however, her comparison of a novel method with her previously constructed manual dictionary that reportedly required around 1,500 person hours is based on an estimate rather than a controlled measurement. We are not aware of previous work comparing development time for named entity taggers in a controlled fashion.

2. METHOD

2.1 Task

Inspired by the Surprise Language Task [5], a "surprise domain" (chosen and annotated without the knowledge of the system developers) was selected: an astronomy data set comprising abstracts of radio astronomical papers was picked, in which the non-standard entity types *instrument name* (names of measurement instruments), *source name* (celestial objects), *source type* (types of objects), and *spectral feature* (spectral lines and their properties) needed to be marked up [1]. Then a group of developers set out to develop three named entity taggers for these named entity classes using three different methods.

2.2 Experimental Setup

To measure the cost in time of developing the three named entity taggers described in this report, we used a Web-based time tracking tool. Participating language engineers and domain experts were asked to bookmark the location of a Web based graphical user interface, and to use it to record briefly their identity and the length and content of each work session. Each time slot was categorised into one of five classes: (I.) System 1: Co-training; (II.) System 2: Active learning; (III.) System 3: Clustering; (IV) Infrastructure; and (V) Communication. The first three of these indicate that time was dedicated to a task specific to one of the three methods covered in this study. The Active Learning effort includes five hours of additional, (inter-)active annotation. The Infrastructure category was

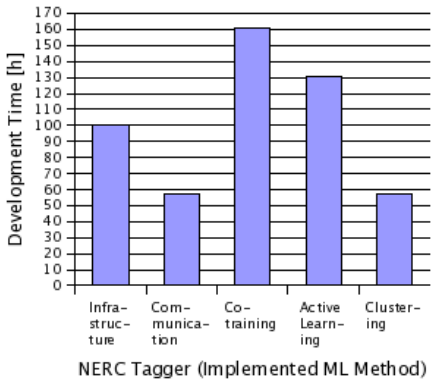

Figure 1: Resource monitoring results.

Method	Development time [h]	F_1-Score
Co-training	319.50	69.06 %
Active learning	**288.00**	**79.50 %**
Clustering	215.00	58.10 %

Table 1: Person hours per method and resulting performance.

used to assign tasks to that relate to the overall setup, such as writing batch scripts to evaluate or convert data-sets. Communication involved attending regular or special-purpose meetings.[1]

3. RESULTS

The resulting time records by category are shown in Figure 1. If the Infrastructure and Communication categories are added to each method's individual development time, we can obtain a conservative estimate of cost per method (Table 1).

4. DISCUSSION AND FUTURE WORK

Individuals developed different recording behaviour: some tracked their work immediately, whereas others preferred to take paper notes and track their time in one "batch" session. Researchers were allowed to record other researchers' time on their behalf, and this was used, for instance using email to ask for time to be accounted when remote access to the intranet (to which the use of the time tracking tool was restricted) was felt to be more cumbersome than sending an email. In our study, Co-training was found to have the highest cost by far; Clustering was found to have the least cost and performance. The differences are not very dramatic in absolute terms, but the evidence overall seems to favour Active Learning. This raises the question whether differences are truly caused by a method or whether they are an artifact of developer experience. Our time monitoring setup can be criticised for its lack of strictness: it does not enforce technically that every minute is really accounted for, because monitoring is a voluntary activity, and while there is no incentive to track time, researchers have many motivations to ignore it: it might be forgotten or neglected due to time pressure. One alternative is automatic time tracking; however, this is difficult to achieve in an environment where researchers have to balance their time between several projects. In addition, there might be ethical implications: for instance, [3] use WinVNC to monitor corporate use of email automatically and without the knowledge of employees. We believe this represents an intrusion of the subjects private sphere, and is illegal without consent in many countries. The time tracking tool used in this study appears to be a practical compromise; it was easy to implement and to use, and a certain lack of accuracy is outweighed by ethical or legal advantages. The results allow a rough quantification of the development cost of the project.[2] If we average the development times per method from Table 1, we find that 35 days are required (development time, not calendar time), amounting to a cost of $8,750 for one tagger.[3] Future work should consider modelling development cost to ultimately allow approximately correct project cost predictions.

[1] For lack of space, cf. [1] for details on the data-set used.

5. SUMMARY AND CONCLUSIONS

We present the first study of (time) resource monitoring for the construction of a set of three named entity taggers for the same task, based on three different, previously published methods—Co-training, Clustering, and Active Learning. The results show that development cost does not differ dramatically across alternative methods, and the fact that Co-training development cost was found higher than Active Learning leads us to the conjecture that differences might be artifacts of developer experience rather than intrinsic advantages or disadvantages of particular methods. We believe further (eventually stricter and more fine-grained) time monitoring experiments will have to be conducted to develop Language Engineering into a discipline that abides by the engineering principle of *development to specification* [2].

Acknowledgements. This study was funded in part by Edinburgh-Stanford Link grant R36759 and DAAD scholarship D/02/01831. Thanks to C. Grover, E. Klein, M. Osborne, B. Webber, and the subjects for this study: B. Alex, M. Becker, S. Dingare, R. Dowsett, B. Hachey, O. Johnson, Y. Krymolowski, R. Mann and M. Nissim.

6. REFERENCES

[1] M. Becker, B. Hachey, B. Alex, and C. Grover. Optimising selective sampling for bootstrapping named entity recognition. In *Proceedings of the ICML 2005 Workshop on Learning with Multiple Views*, Bonn, Germany, 2005.

[2] H. Czichos, M. Hennecke, and Akademischer Verein Hütte e. V. Berlin, editors. *HÜTTE – Das Ingenieurwissen*. Springer, Berlin, 32nd edition, 2004.

[3] T. W. Jackson, R. Dawson, and D. Wilson. Understanding email interaction increases organizational productivity. *Commun. ACM*, 46(8):80–84, 2003.

[4] C. F. Kemerer. An empirical validation of software cost estimation models. *Communications of the ACM*, 30(5):416–429, 1987.

[5] D. W. Oard. The surprise language exercises. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(2):79–84, 2003.

[6] E. Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI'93)*, pages 811–816. AAAI, AAAI Press and MIT Press, 1993.

[2] Assuming equal staff salaries for language engineers and domain experts of $75,000 per year (taken from a real job ad), and assuming 300 workdays per year, the staff cost per capita and day amounts to $250. The 64 person days consumed by the project then amount to $16,000.

[3] This calculation does not model project delays caused by a team member waiting for another developer to complete a task.