4REAL Workshop:
Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language

Workshop Programme

09:00 – 10:30 Reproducibility

Luís Gomes, Gertjan van Noord and António Branco, Steve Neale, *Seeking to Reproduce "Easy Domain Adaptation"*

Kevin Cohen, Jingbo Xia, Christophe Roeder and Lawrence Hunter, *Reproducibility in Natural Language Processing: A Case Study of two R Libraries for Mining PubMed/MEDLINE*

Filip Graliński, Rafał Jaworski, Łukasz Borchmann and Piotr Wierzchoń, *Gonito.net - Open Platform for Research Competition, Cooperation and Reproducibility*

10:30 – 11:00 Coffee break

11:00 – 12:00 Citation

Jozef Milšutka, Ondřej Košarko and Amir Kamran, *SHORTREF.ORG - Making URLs Easy-to-Cite*

Gil Francopoulo, Joseph Mariani and Patrick Paroubek, *Linking Language Resources and NLP Papers*

12:00 – 13:00 Round table

Reproducibility in Language Science and Technology: Ready for the Integrity Debate?
Chair: António Branco

Editors

António Branco	University of Lisbon
Nicoletta Calzolari	ILC-CNR / ELRA
Khalid Choukri	ELDA

Workshop Organizers/Organizing Committee

António Branco	University of Lisbon
Nicoletta Calzolari	ILC-CNR / ELRA
Khalid Choukri	ELDA

Workshop Programme Committee

António Branco	University of Lisbon
Iryna Gurevych	Universität Darmstadt
Isabel Trancoso	INESC-ID / IST, University of Lisbon
Joseph Mariani	CNRS/LIMSI
Justus Roux	NWVU
Khalid Choukri	ELDA
Maria Gavrilidou	ILSP
Marko Grobelnik	Jozef Stefan Institute
Marko Tadic	University of Zagreb
Mike Rosner	University of Malta
Nicoletta Calzolari	ILC-CNR/ELRA
Nick Campbell	Trinity College Dublin
Senja Pollak	Jozef Stefan Institute
Stelios Piperidis	ILSP
Steven Krauwer	University of Utrecht
Thierry Declerck	DFKI
Torsten Zesch	University of Duisburg-Essen
Yohei Murakami	Language Grid Japan

Table of contents

Author Index

Introduction

The discussion on research integrity has grown in importance as the resources allocated to and societal impact of scientific activities have been expanding (e.g. Stodden, 2013, Aarts *et al*., 2015), to the point that it has recently crossed the borders of the research world and made its appearance in important mass media and was brought to the attention of the general public (e.g. Nail and Gautam, 2011, Zimmer, 2012, Begley and Sharon 2012, The Economist, 2013).

The immediate motivation for this increased interest is to be found in a number of factors, including the realization that for some published results, their replication is not being obtained (e.g. Prinz *et al*., 2011; Belgley and Ellis, 2012); that there may be problems with the commonly accepted reviewing procedures, where deliberately falsified submissions, with fabricated errors and fake authors, get accepted even in respectable journals (e.g. Bohannon, 2013); that the expectation of researchers vis a vis misconduct, as revealed in inquiries to scientists on questionable practices, scores higher than one might expect or would be ready to accept (e.g. Fanelli, 2009); among several others.

Doing justice to and building on the inherent ethos of scientific inquiry, this issue has been under thorough inquiry leading to a scrutiny on its possible immediate causes and underlying factors, and to initiatives to respond to its challenge, namely by the setting up of dedicated conferences (e.g. WCRI – World Conference on Research Integrity), dedicated journals (e.g. RIPR – Research Integrity and Peer review), support platforms (e.g. COS – Center for Open Science), revised and more stringent procedures (e.g. Nature, 2013), batch replication studies (e.g. Aarts *et al*., 2015), investigations on misconduct (e.g. Hvistendahl, 2013), etc.

This workshop seeks to foster the discussion and the advancement on a topic that has been so far given insufficient attention in the research area of language processing tools and resources (Branco, 2013, Fokkens *et al*., 2013) and that has been an important topic emerging in other scientific areas. That is the topic of the reproducibility of research results and the citation of resources, and its impact on research integrity.

We invited submissions of articles that presented pioneering cases, either with positive or negative results, of actual replication exercises of previous published results in our area. We were interested also in articles discussing the challenges, the risk factors, the procedures, etc. specific to our area or that should be adopted, or adapted from other neighboring areas, possibly including of course the new risks raised by the replication articles themselves and their own integrity, in view of the preservation of the reputation of colleagues and works whose results are reported has having been replicated, etc.

By the same token, this workshop was interested also on articles addressing methodologies for monitoring, maintaining or improving citation of language resources and tools and to assess the importance of data citation for research integrity and for the advancement of natural language science and technology.

The present volume gathers the papers that were selected for presentation and publication after having received the sufficiently positive evaluation by three reviewers from the workshop's program committee.

We hope this workshop, collocated with the LREC 2016 conference, will help to open and foster the discussion on research results reproducibility and resources citation in the domain of science and technology of language.

18 April 2016

António Branco, Nicoletta Calzolari and Khalid Choukri

References:

Aarts, et al., 2015, “Estimating the Reproducibility of Psychological Science”, *Science*.
The Economist, 2013, "Unreliable Research: Trouble at the Lab", *The Economist*, October 19, 2013, online.
Begley, 2012, "In Cancer Science, Many “Discoveries” don’t hold up", *Reuters*, March 28th, online.
Begley and Ellis, 2012, "Drug Development: Raise Standards for Preclinical Cancer Research", *Nature*.
Bohannon, 2013, "Who’s Afraid of Peer Review?", *Science*.
Branco, 2013, "Reliability and Meta-reliability of Language Resources: Ready to initiate the Integrity Debate?", In *Proceedings of The 12th Workshop on Treebanks and Linguistic Theories* (TLT12).
COS, Centre for Open Science.
Fanelli, 2009, "How Many Scientists Fabricate and Falsify Research? A Systematic Review and Meta-Analysis of Survey Data", *PL OS ONE*.
Fokkens, van Erp, Postma, Pedersen, Vossen and Freire, 2013, "Offspring from Reproduction Problems: What Replication Failure Teaches US", In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (ACL2013).
Hvistendahl, 2013, “China’s Publication Bazaar”, *Science*.
Nail, 2011, "Scientists’ Elusive Goal: Reproducing Study Results", *The Wall Street Journal*.
Nature, 2013, “Announcement: Reducing our Irreproducibility”, *Nature*, Editorial.
Prinz, Sclange and Asadullah, 2011, "Believe it or not: How much can We Rely on Published Data on Potential Drug Targets?", *Nature Reviews Drug Discovery* 10, 712.
RIPR, Research Integrity and Peer Review.
Stodden, 2013, "Resolving Irreproducibility in Empirical and Computational Research", *IMS Bulletin Online*.
WCRI, World Conference on Research Integrity.
Zimmer, 2012, "A Sharp Rise in Retractions Prompts Calls for Reform", *The New York Times*.

Seeking to Reproduce "Easy Domain Adaptation"

Luís Gomes*, **Gertjan van Noord**†, **António Branco***, **Steven Neale***
*NLX – Natural Language and Speech Group
Faculdade de Ciências, Universidade de Lisboa, Portugal
{luis.gomes,steven.neale,antonio.branco}@di.fc.ul.pt

†University of Groningen, Netherlands
g.j.m.van.noord@rug.nl

Abstract

The *frustratingly easy domain adaptation* technique proposed by Daumé III (2007) is simple, easy to implement, and reported to be very successful in a range of NLP tasks (named entity recognition, part-of-speech tagging, and shallow parsing), giving us high hopes of successfully replicating and applying it to an English↔Portuguese hybrid machine translation system. While our hopes became 'frustration' in one translation direction – as the results obtained with the domain-adapted model do not improve upon the in-domain baseline model – our results are more encouraging in the opposite direction. This paper describes our replication of the technique and our application of it to machine translation, and offers a discussion on possible reasons for our mixed success in doing so.

1. Introduction

Frustratingly easy domain adaptation (EasyAdapt) is a technique put forward by Daumé III (2007) that allows learning algorithms that perform well across multiple domains to be easily developed. The technique is based on the principle of augmenting features from source language text in one domain – for which there might be more training data available – with features from text in a second, target domain in order that this domain is represented within a larger quantity of input data that can be fed to a learning algorithm. As well as purportedly being 'incredibly' easy to implement, the technique is shown to outperform existing results in a number of NLP tasks, including named entity recognition (NER), part-of-speech (POS) tagging, and shallow parsing.

Against this backdrop, we had high hopes of replicating the EasyAdapt technique and implementing it in the hybrid tree-to-tree machine translation (MT) system (Silva et al., 2015; Dušek et al., 2015) we have been developing as part of the QTLeap project[1]. While our system – based on the hybrid MT framework *TectoMT* (Zabokrtsky et al., 2008) – had been primarily trained on the much larger and broader-domained Europarl (EP) corpus (Silva et al., 2015; Dušek et al., 2015), we recently constructed a smaller, in-domain (IT) corpus of parallel questions and answers taken from real users' interactions with an information technology company's question answering (QA) system. Having initially obtained slightly improved results using this small in-domain corpus to train our MT system in the English↔Portuguese directions, we saw a potential benefit to replicating the EasyAdapt model and using it to produce larger, targeted training data encompassing both corpora.

In this paper, we report our results from replicating the EasyAdapt technique and applying it to the maximum entropy (MaxEnt) transfer models on which our hybrid tree-to-tree MT system is based, thus making use of an augmentation of features from both the larger, broader domain (EP) corpus and the smaller, in-domain (IT) corpus. Despite our initial results being slightly better when training our system on the IT corpus than with the much larger EP corpus, we were left frustrated after seeing encouraging results in only one translation direction when combining the two using the EasyAdapt model. As well as reporting our results using the EasyAdapt model, we also describe why – despite having been shown by Daumé III (2007) to be a successful technique for a range of NLP tasks – our attempts to replicate the model for our MT system did not lead to similarly improved results in both translation directions.

2. Hybrid Tree-to-Tree MT

The QTLeap project explores deep language engineering approaches to improving the quality of machine translation, and involves the creation of MT systems for seven languages paired with English: Basque, Bulgarian, Czech, Dutch, German, Portuguese, and Spanish. The systems are being used to translate the questions posed by users of an IT company's interactive QA system from their native language into English (the primary language of the QA system's database), using these machine-translations to retrieve the most similar questions and their respective answers from said database. The retrieved answers are then machine-translated back into the user's native language – if they have not already been translated before – to complete the cycle and provide users with an answer to their initial question.

For English↔Portuguese, our translation pipeline is based on the tectogrammatical hybrid-MT framework *TectoMT* (Zabokrtsky et al., 2008) and follows a classical analysis→transfer→synthesis structure with the transfer taking place at a deep syntactic (tectogrammatical) level of representation.

2.1. Analysis

The analysis of input sentences is performed in two stages: the first stage takes the raw string representation and produces a surface-level analytical tree (a-tree) representation; the second stage takes the a-tree and produces a deeper tectogrammatical tree representation (t-tree). Figure 1 shows

[1] http://qtleap.eu

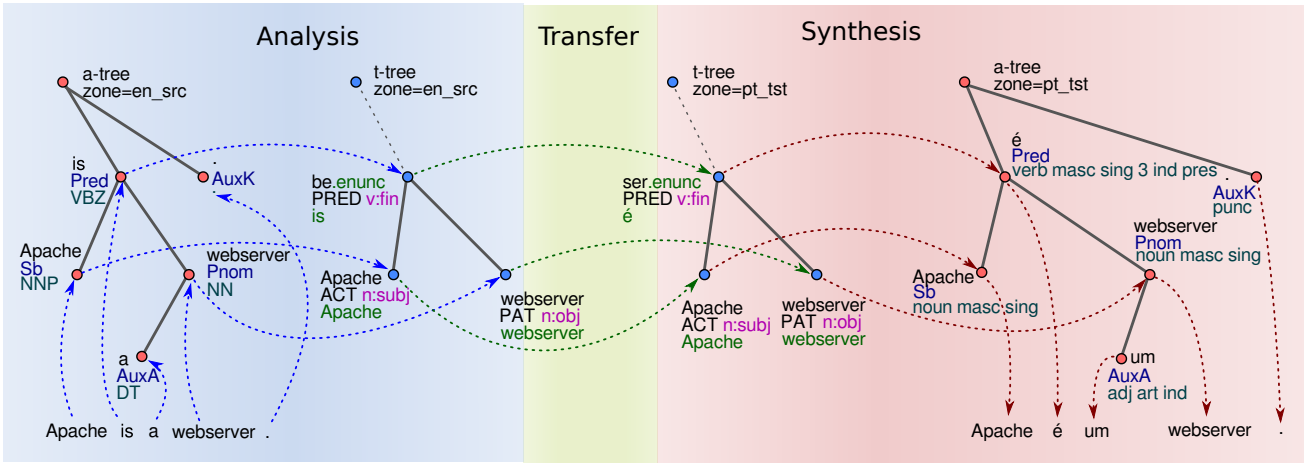


Figure 1: Trees at several stages of the analysis-transfer-synthesis of the pipeline.

the a-trees and t-trees of a short sentence as it is translated from English to Portuguese.

In the surface-level analytical tree representation of a sentence (a-tree), each word is represented by a node and the edges represent syntactic dependency relations. Nodes have several attributes, the most relevant being the lemma, part-of-speech tag, and morphosyntactic features (gender, number, tense, person, etc). By contrast, in the tectogrammatical tree representation (t-tree) only content words (nouns, pronouns, main verbs, adjectives and adverbs) are represented as nodes, but the t-tree may also contain nodes that have no corresponding word in the original sentence, such as nodes representing pro-dropped pronouns.

2.1.1. Surface-level analysis

For the initial surface-level analysis we use LX-Suite (Branco and Silva, 2006), a set of shallow processing tools for Portuguese that includes a sentence segmenter, a tokenizer, a PoS tagger, a morphological analyser and a dependency parser. All of the LX-Suite components have state-of-the-art performance. The trees produced by the dependecy parser are converted into the Universal Stanford Dependencies tagset (USD) proposed by de Marneffe et al. (2014). Additionally, the part-of-speech and morphological feature tags are converted into the Interset tagset (Zeman, 2008).

2.1.2. Deeper analysis

The second analysis stage transforms the surface-level a-trees into deeper t-trees. This transformation is purely rule-based; some rules are language-specific, others are language-independent. The two most important transformations are: (1) drop nodes corresponding to non-content words (articles, prepositions, auxiliary verbs, etc) and (2) add nodes for pro-dropped pronouns. Because all TectoMT-based translation pipelines adopt a universal representation for a-trees (USD and Interset), the language-independent rules are shareable across pipelines, reducing the amount of work needed to add support for new languages.

2.2. Transfer

The transfer step is statistical and is responsible for transforming a source-language t-tree into a target-language t-tree. It is assumed that the source and target t-trees are isomorphic, which is true most of the time, given that at the tectogrammatical representation level, most language-dependent features have been abstracted away. Thus, the transformation of a source-language t-tree into a target-language t-tree is done by statistically transferring node attributes (t-lemmas and formemes) and then reordering nodes as needed to meet the target-language word ordering rules. Some reorderings are encoded in formemes, as for example *adj:prenon* and *adj:postnom*, which represent prenominal and postnominal adjectives respectively.

2.2.1. Transfer Models

The transfer models are multi-label classifiers that predict an attribute (t-lemma or formeme) of each target-language t-node given as input a set of attributes of the corresponding source-language t-node and its immediate neighbours (parent, siblings and children). There are separate models for predicting t-lemmas and formemes, but other than the different output labels, the input feature sets are identical. Two kinds of statistical models are employed and interpolated: (1) a *static* model that predicts output t-lemmas (or formeme) based solely on the source-language t-lemma (or formeme), i.e. without taking into account any other contextual feature, and (2) a *MaxEnt* model[2] that takes all contextual features into account.

2.3. Synthesis

The synthesis step of the pipeline is rule-based and relies on two pre-existing tools for Portuguese synthesis: a verbal conjugator (Branco and Nunes, 2012) and a nominal inflector (Martins, 2006). Besides these synthesis tools, there are rules for adding auxiliary verbs, articles and prepositions as needed to transform the deep tectogrammatical representation into a surface-level tree representation, which is then converted into the final string representation by concatenating nodes (words) in depth-first left-to-right tree-traversal ordering (adding spaces as needed).

[2]there is one MaxEnt model for each distinct source-language t-lemma (or formeme) so, in fact, we have an ensemble of MaxEnt models

3. Frustratingly Easy Domain Adaptation

The 'frustratingly easy domain adaptation' (EasyAdapt) technique (Daumé III, 2007) is a simple feature augmentation technique that can be used in combination with many learning algorithms. The application of EasyAdapt for various NLP tasks, including Named Entity Recognition, Part-of-Speech Tagging, and Shallow Parsing was reported as successfull. Even if EasyAdapt is not directly applicable to the models typically used in Statistical Machine Translation, a similar approach has been shown to improve results for translation as well (Clark et al., 2012).

Although EasyAdapt has been developed in the context of domain adaptation, it is best described as a very simple, yet effective, multi-domain learning technique (Joshi et al., 2012). In EasyAdapt, each input feature is augmented with domain specific versions of it. If we have data from K domains, the augmented feature space will consist of $K + 1$ copies of the original feature space. Each training/testing instance is associated with a particular domain, and therefore two versions of each feature are present for a given instance: the original, general, version and the domain specific version.

The classifier may learn that a specific feature is always important, regardless of the domain (and thus it will rely more on the general version of the feature), or it may learn that a specific feature is relevant only for particular domain(s) and thus rely more on the relevant domain specific features. As a result, we obtain a single model which encodes both generic properties of the task as well as domain specific preferences.

We implemented EasyAdapt in our MaxEnt transfer models by adding, for each original feature f, a feature f_d if the training/testing instance is from domain d. In the experiments below, there are only two domains, the IT domain, which we regard as in-domain for the translation system, and the EP domain, which is out-of-domain for our translation system. [3].

4. Experiments

We performed a total of 24 experiments, evaluating four different models on three testsets (models and testsets outlined below) and in both translation directions – English→Portuguese and Portuguese→English.

4.1. Models

The smaller, in-domain parallel corpus we created for training the transfer models comprises 2000 questions and 2000 answers collected from real user interactions with the QA-based chat system used by an information technology company to provide technical assistance to its customers. The out-of-domain corpus is the English and Portuguese-aligned version of Europarl (Koehn, 2005).

From these two corpora, we created four models: **IT** (trained with what we consider to be our in-domain data only, the 4000 sentences from the user interactions with the QA system), **EP** (trained with what we consider to be our out-of-domain data only, the Europarl corpus), **IT+EP** (a trivially domain-adapted model obtained by concatenating both the IT and the EP corpora), and finally the **EasyAdapt** model (a domain-adapted model created by using the EasyAdapt technique to combine features from both corpora).

4.2. Testsets

We have used three testsets for evaluation: **IT** (an in-domain testset composed of 1000 questions and 1000 answers collected from the same real user interactions with the QA-based chat system as the previously described model, but with no overlap with the corpus used for training), **News** (an out-of-domain testset with 604 sentences from the news domain, created by manually translating a subset of the testset used in the WMT12 tasks[4] into Portuguese in the context of the QTLeap project), and **EP** (the first 1000 parallel sentences in English and Portuguese from the Europarl corpus).

Note that the **News** testset is from a different domain than either of the other two corpora (IT and EP) used for training – we wanted to experiment with this additional testset to see whether or not the EasyAdapt model is more general than the model obtained by simply concatenating both corpora (IT+EP).

4.3. Results

Tables 1 and 2 show the BLEU (Papineni et al., 2002) and NIST (Doddington, 2002) scores obtained with the four models on the three testsets for the English→Portuguese and Portuguese→English translation directions respectively. Frustratingly, the EasyAdapt model did not outperform the baseline in-domain model (IT) on the in-domain (IT) testset for English→Portuguese. However, the EasyAdapt model was the best performing model in the Portuguese→English direction. In this context, our initial goal of improving in-domain translation by learning from a larger out-of-domain corpus augmented with features from a smaller, targeted in-domain corpus using the EasyAdapt model has been met with mixed success.

For the better performing direction of Portuguese→English, the scores obtained using the EasyAdapt model outperform other models on all but the EP testset, for which they are only slightly behind. This suggests that in the scenario that we need to use a single model to translate two domains instead of a separate model for each domain – to ease memory concerns, perhaps – the EasyAdapt model would likely be a much better option than simply concatenating the corpora from both domains. Furthermore, the EasyAdapt model is the best performing model when translating texts from a third (News) domain.

5. Discussion

Although the EasyAdapt model was effective in the Portuguese→English direction, we were disappointed that the same good results were not obtained for in-domain translation in the English→Portuguese direction. Considering possible reasons for this, we note that the development

[3] Below, we also apply our models to a third domain, News, but since we do not train on that domain, there is no point in having News-specific features

[4] `http://www.statmt.org/wmt12/test.tgz`

of our hybrid MT system has so far been more heavily concentrated on the English→Portuguese direction given that – as described in section 2. – this is the direction whose translation will be presented to end users. As a result, the system was weaker in the Portuguese→English direction to begin with.

With this in mind, we expect that there is more room for the EasyAdapt model to impact on results in a positive manner in the Portuguese→English than in the English→Portuguese translation direction, and that this is why we see improvements in translation quality when using the model in this direction. This is also likely when we consider that the kinds of tasks for which Daumé III (2007) reported improved performance – pos tagging, shallow parsing etc. – are surface-level in nature, as are both the shallow-processing tasks used in the analysis phase and the rule-based components used in the synthesis phase of the hybrid MT system.

Considering the synthesis steps in particular, the fact that the Portuguese components used in the English→Portuguese direction are more matured and have received more attention than their Portuguese→English counterparts may simply mean that these components already perform well enough that no real improvements can be seen from using the EasyAdapt model, in contrast to the equivalent components in the opposite direction which are less mature and therefore improved by adopting the EasyAdapt model.

Model	Testset					
	IT		News		EP	
	BLEU	NIST	BLEU	NIST	BLEU	NIST
IT	**22.81**	**6.47**	4.10	3.21	4.25	2.72
EP	18.73	5.60	8.03	**4.46**	**8.00**	4.39
IT+EP	21.25	6.09	7.84	4.43	7.89	4.36
EasyAdapt	22.63	6.44	**8.13**	4.40	7.82	**4.43**

Table 1: BLEU and NIST scores obtained with four transfer models (rows) in three different domain testsets (columns) for the English→Portuguese direction.

Model	Testset					
	IT		News		EP	
	BLEU	NIST	BLEU	NIST	BLEU	NIST
IT	13.78	4.97	2.77	2.90	2.41	2.50
EP	12.24	4.43	6.57	4.13	**7.25**	4.24
IT+EP	13.30	4.78	6.46	4.11	7.09	4.18
EasyAdapt	**14.13**	**5.13**	**6.81**	**4.18**	7.13	**4.25**

Table 2: BLEU and NIST scores obtained with four transfer models (rows) in three different domain testsets (columns) for the Portuguese→English direction.

6. Conclusions

We have presented the results of our replication of the EasyAdapt (*frustatingly easy domain adaptaion*) technique and our integration of it into an English↔Portuguese hybrid machine translation system. We had high hopes that by replicating the technique, we would be able to combine features from the large out-of-domain (EP) corpus we had previously used to train our system with features from a small in-domain (IT) corpus constructed within the scope of the QTLeap project and see improved results by feeding this combination of features to our maxent-based transfer models during the training of the system.

Our efforts to reproduce the improvements of the EasyAdapt technique reported by Daumé III (2007) have been of mixed success in the context of machine translation. While we were able to improve the Portuguese→English translation of in-domain texts using the EasyAdapt technique compared to the in-domain trained baseline, the EasyAdapt model did not outperform the in-domain trained baseline in the English→Portuguese direction, which is currently our best performing of the two directions. Among other possible reasons for this, it may simply be the case that as Portuguese→English is our weaker translation direction, the EasyAdapt model has more room to make an impact on translations and less so in the more matured and refined pipeline for the English→Portuguese direction.

The EasyAdapt technique was reported to lead to better results in a number of NLP tasks by preparing domain-adapted training data (Daumé III, 2007) but we have found it difficult to fully reproduce that success across the board in the machine translation context. These results highlight the importance of replicating techniques in different contexts to truly assess their suitability to and reproducibility of results across different scenarios.

7. Acknowledgements

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7) under grant agreement nº 610516 (project QTLeap), and from the Portuguese Foundation for Science and Technology (FCT) under grant PTDC/EEI-SII/1940/2012 (project DP4LT).

8. References

Branco, A. and Nunes, F. (2012). Verb analysis in a highly inflective language with an mff algorithm. In *Computational Processing of the Portuguese Language*, pages 1–11. Springer.

Branco, A. and Silva, J. (2006). A suite of shallow processing tools for Portuguese: LX-Suite. In *Proceedings of the 11th European Chapter of the Association for Computational Linguistics*, pages 179–182.

Clark, J., Lavie, A., and Dyer, C. (2012). One system, many domains: Open-domain statistical machine translation via feature augmentation. In *AMTA 2012, Conference of the Association for Machine Translation in the Americas*, San Diego, October 28 – November 1.

Daumé III, H. (2007). Frustratingly easy domain adaptation. *ACL 2007*, page 256.

de Marneffe, M.-C., Silveira, N., Dozat, T., Haverinen, K., Ginter, F., Nivre, J., and Manning, C. (2014). Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the 9th Language Resources and Evaluation Conference*.

Doddington, G. (2002). Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc.

Dušek, O., Gomes, L., Novák, M., Popel, M., and Rosa, R. (2015). New language pairs in tectomt. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 98–104.

Joshi, M., Cohen, W. W., Dredze, M., and Rosé, C. P. (2012). Multi-domain learning: When do domains matter? In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1302–1312, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5.

Martins, P. (2006). LX-Inflector: Implementation Report and User Manual, University of Lisbon. Technical report, TagShare Project.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Silva, J., Rodrigues, J., Gomes, L., and Branco, A. (2015). Bootstrapping a hybrid deep MT system. In *Proceedings of the Fourth Workshop on Hybrid Approaches to Translation (HyTra)*, pages 1–5, Beijing, China, July. Association for Computational Linguistics.

Zabokrtsky, Z., Ptacek, J., and Pajas, P. (2008). TectoMT: Highly modular MT system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170, Columbus, Ohio, June. Association for Computational Linguistics.

Zeman, D. (2008). Reusable tagset conversion using tagset drivers. In *LREC*.

Reproducibility in Natural Language Processing: A Case Study of Two R Libraries for Mining PubMed/MEDLINE

K. Bretonnel Cohen[1], Jingbo Xia[2], Christophe Roeder, and Lawrence E. Hunter[1]
[1]Biomedical Text Mining Group
Computational Bioscience Program,
University of Colorado School of Medicine
[2]Department of Bio-statistics,
College of Informatics,
Hubei Key Laboratory of Agricultural Bioinformatics,
Huazhong Agricultural University

Abstract

There is currently a crisis in science related to highly publicized failures to reproduce large numbers of published studies. The current work proposes, by way of case studies, a methodology for moving the study of reproducibility in computational work to a full stage beyond that of earlier work. Specifically, it presents a case study in attempting to reproduce the reports of two R libraries for doing text mining of the PubMed/MEDLINE repository of scientific publications. The main findings are that a rational paradigm for reproduction of natural language processing papers can be established; the advertised functionality was difficult, but not impossible, to reproduce; and reproducibility studies can produce additional insights into the functioning of the published system. Additionally, the work on reproducibility lead to the production of novel user-centered documentation that has been accessed 260 times since its publication—an average of once a day per library.

Keywords: reproducibility, natural language processing, PubMed/MEDLINE

1. Introduction

The general crisis of (non-)reproducibility in science extends into natural language processing research (Pedersen, 2008; Branco, 2012; Fokkens et al., 2013). The authors of this paper are well aware that we ourselves have made software publicly available that no longer runs, or no longer functions completely, or is no longer available, despite having published URLs for it. The goal of this paper is to help to establish a methodology for exploring issues of reproducibility in the field.

It is not hyperbole to say that there is a crisis in science related to highly publicized failures to reproduce large numbers of published studies. The phenomenon has been observed in fields as diverse as computer science (Collberg et al., 2014b; Collberg et al., 2014a; Proebsting and Warren, 2015), psychology (Collaboration and others, 2012), signal processing (Kovacevic, 2007; Vandewalle et al., 2009), cancer biology (Barrows et al., 2010; Prinz et al., 2011), medicine (Begley and Ellis, 2012; Mobley et al., 2013), and biomedical research in general (Collins and Tabak, 2014), with implications even for fields that bridge the social and humanistic sciences, such as classic linguistic field research (Bisang, 2011; Berez, 2015).

Does this crisis really extend to natural language processing? There is some reason to think that it does not. A number of libraries, executables, and architectures for natural language processing have been published on and subsequently used extensively by large numbers of other researchers. These artifacts have been subjected to extensive "testing" in the form of their uses "in the wild," and some of them have held up to intensive use. However, these encouraging facts might not be representative of the state of the natural language processing software ecosystem as a whole. Impressionistically, in addition to this set of highly successful natural language processing distributions, there are myriad applications reported in the literature that turn out to be uncompilable, unusable, unobtainable, or otherwise not reproducible (Pedersen, 2008; Poprat et al., 2008). This paper attempts to move the field beyond those impressionistic observations to a rational approach to assessing reproducibility in natural language processing. We report on our experiences with the pubmed.mineR (Rani et al., 2015) and rentrez libraries for the R programming language. These libraries provide a number of affordances for doing text mining of the PubMed/MEDLINE repository of biomedical publications. PubMed/MEDLINE is a prominent part of the biomedical research milieu, with 23 million entries and new ones being added at a rate of 2,700 per day. Text mining, especially of the PubMed/MEDLINE repository, is a booming field in the bioscience and bioinformatics communities. Pubmed.mineR and rentrez attempt to facilitate that work with the R language. R provides an extensive range of affordances for statistics and graphing, and is one of the fastest-growing languages in the world (Ihaka and Gentleman, 1996).

To see the motivation for the approach that we describe here, consider Figure 1. It shows the increase in processing time as a popular publicly available language processing API is given increasingly large inputs. This is one of the systems mentioned above, and it has been used extensively. Note that processing times increase logarithmically (correlation coefficient of fit to log model = 0.80) up to about 18,000 words, followed soon by a program crash at 19,000 tokens. Under what conditions can we say that the many publications on this system's performance are reproducible? At the most, we can assume them to be reproducible only up to about 18,000 words of input (assuming memory and other configuration similar to the machine that

we used at the time). Input size under which the reported performance numbers hold is not something that is reported or (as far as the authors can tell) considered in those publications, but the data reported in Figure 1 suggests that the reported performance is not, in fact, reproducible for all possible inputs, and the logarithmic increase in processing times suggests that as the memory on the machine reaches its limits, the application is not robust in the face of phenomena like swapping memory to disk.

1.1. Problems of reproducibility in natural language processing

A number of factors conspire to make reproducibility in any traditional sense difficult to impossible in the domain of natural language processing.

- The data is often not available (Branco, 2012). In some cases, the shared task model has made great progress towards addressing this issue. In other cases, such as natural language processing in the medical, intelligence, and law enforcement domains, the problem of unavailability of data will probably never be addressed in such a way as to facilitate reproducibility.
- Natural language processing research is primarily published in conference proceedings, not journals. Because conference papers routinely have page limits, there is typically not enough space to give all information on the methodology that would be necessary to replicate the work (see, for example, (Fokkens et al., 2013)).
- There is little or no tradition in the community of publishing reproduction attempts—the bias is strongly in favor of novel methods (Fokkens et al., 2013).

1.1.1. The reproducibility hierarchy of needs

There are a number of conditions that must be met in order to reproduce a study in natural language processing. These form a hierarchy—if the most basic conditions cannot be met, then the higher ones cannot, either. We consider here a typical natural language processing paper reporting a new tool and/or method.

1. Availability: the system must be available, or there must be sufficient detail available to reconstruct the system, exactly.
2. Builds: the code must build.
3. Runs: The built code must run.
4. Evaluation: it must be possible to run on the same data and measure the output using the same implementation of the same scoring metric.

Most of the sociology of natural language processing militates against all steps in the hierarchy being met. Limits on conference paper lengths assure that there will rarely be enough information available in the methodology section to reconstruct the system. GitHub and similar distribution mechanisms have, of course, made it easier to distribute versioned code, but many people still report not being able to find code, not being able to remember how to build it, etc. Maven has made progress in ensuring that build processes are repeatable, but most projects in NLP are not distributed as Maven projects, which in any case are not appropriate for every language and architecture used in NLP research. Even given a built program, it may not run, e.g. due to undocumented platform dependencies, configuration files, input requirements, memory requirements, processor requirements, graphics card requirements, etc.

An experiment by (Collberg et al., 2014a) looked at levels 1 and 2 of this hierarchy in a study of systems reported at computer science conferences. The results are discussed elsewhere in this paper. The work resulted in a principled approach to evaluating the extent of buildability in CS research reproduction. That was clearly difficult to do in a reasonably methodologically sound way. The work reported here attempts to go to the next level–evaluating not buildability, but executability–and it is not immediately clear what that methodology should be. This paper is a step towards developing such a methodology, or a framework for developing such a methodology. The novelty of the work reported here is that it takes the effort to level 3—that is, executability. More specifically, it explores the possibilities for working at level 3, and shows some results for work of that nature.

1.1.2. An expanded conception of reproducibility

Collberg et al. (Collberg et al., 2014a) suggest that in the context of computer science research, the notion of reproducibility—defined by them as *the independent confirmation of a scientific hypothesis through reproduction by an independent researcher/lab*—can usefully be replaced by the concept of *repeatability.* In particular, they define three types of what they call *weak repeatability.* The highest level is the ability of a system to be acquired, and then built in 30 minutes or fewer. The next level is the ability of a system to be acquired, and then built, regardless of the time required to do so. The lowest level is the ability of a system to be acquired, and then either built, regardless of the time required to do so, *or* the original author's insistence that the code would build, if only enough of an effort were made.

Code available and builds within 30 minutes	32.3%
Code available and builds	48.3%
Either code builds, or original authors insist that it would	54.0%

Table 1: Summary of results on 402 papers whose results were backed by code, from (Collberg et al., 2014a).

This notion of weak reproducibility is demonstrably useful; Table 1.1.2. shows how effective it was in quantifiying reproducibility issues in computer science. However, as the authors point out, it leaves out crucial elements of reproducibility. For one thing, it does not take versioning into consideration: assuming that code is available and builds, can we necessarily assume that it is the same version as the code that was used in the paper? For another, the defini-

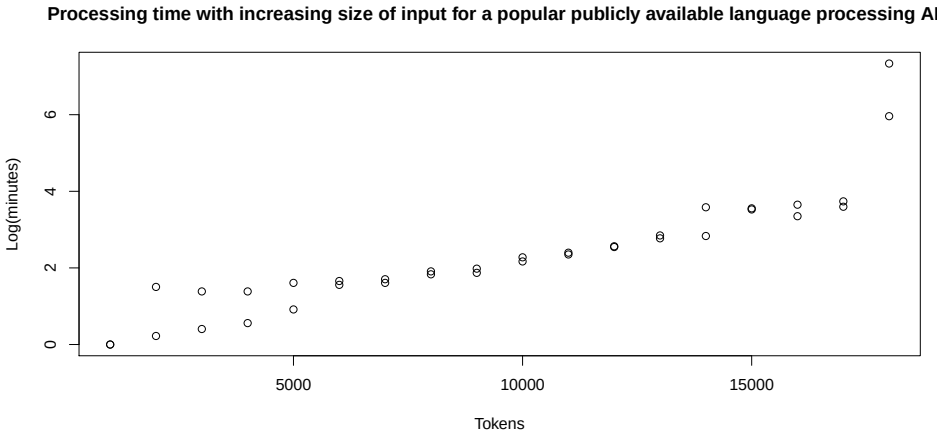


Figure 1: Processing time with increasing size of input for a popular publicly available natural language processing API. Processing time increases logartithmically with input size until 18,000 tokens, and then the program crashes. The experiment was run twice, resulting in two different run times at most input sizes.

tion does not take into consideration what (Collberg et al., 2014a) call *executability:* will the code not just build, but run? For example, even examples from papers don't always work as advertised. We suggest here other work that can be useful in evaluating the claims of a paper. The work reported here tries to tackle the executaility issue specifically. We suggest here some more things to think about:

Processing time: it can be revealing to measure processing times as increasingly large data sets are treated. For example, we found one widely used system that showed a linear increase in processing time with input text size until at some (repeatable) input size the processing time began increasing rapidly, and then (with more increases in input size) the system crashed.

Validating through debugging output: we found one library that produced debugging output that could clearly be demonstrated to be wrong with simple UNIX commands.

Metamorphic testing: natural language processing applications are obvious candidates for metamorphic testing (defined below, in the *Methods* section).

1.2. Related literature

(Collberg et al., 2014a) reviews two previous studies of code-sharing in computer science research. (Kovacevic, 2007) began with 15 IEEE papers and evaluated the presence of proofs, availability of code, and availability of data. They found that all papers presented proofs, none of the papers made code available, and 33% of papers were based on data that was available (probably due to the wide use of publicly available data sets in that field). In our hierarchy of needs, this would be level 1. (Vandewalle et al., 2009) looked at 134 IEEE papers in terms of availability of code and of data, and found that code was available in 9% of cases and data was available in 33% of cases (same field). Again, this corresponds to level 1 of the hierarchy of needs. (Collberg et al., 2014a), discussed elsewhere in this paper, took the work to level 2; this paper advances to level 3, or executability.

2. Materials and methods

Two R libraries for text mining from PubMed/MEDLINE, the primary repository for biomedical publications, were selected for the case study. They are interesting case studies in that they allow the examination of what we are calling level 3 of the hierarchy. Since we know in advance, due to their availability on CRAN, that they are available and they build, they allow us to take the next step of studying their run-time behaviors. They were also selected due to their intended domain of application. While many systems that are reported in the general natural language processing literature are avowedly research systems, and it could be argued that their run-time characteristics were not a focus of the research, the situation is different in biomedical natural language processing. In this specialized domain, the stated purpose of the work is often not research per se, but the goal of providing a working tool to biologists or physicians (Hirschman et al., 2007). Thus, investigations of reproducibility at the level of run-time behavior are clearly relevant in biomedical natural language processing.

2.1. Pubmed.mineR

Pubmed.mineR is a library for doing text mining from the PubMed/MEDLINE collection of documents. PubMed/MEDLINE contains references to about 23 million articles in the domain of biomedical science, broadly construed. Pubmed.mineR provides a clean interface to named entity recognition and normalization web services provided by the National Center for Biotechnology Information via the PubTator application (Wei et al., 2013). Pubmed.mineR was released with documentation for the various and sundry methods that it provides, but no manual.

Two tests are designed to evaluate the performance of pubmed.mineR package. In Test 1, we built four document collections of different sizes and tested the speed of named entity recognition and normalization by using the package. In Test 2, we examined the stability of performance by run-

ning 10 iterations of the largest document set.
All of the experiments were carried out on a Mac desktop with installation of Mac OS X (version 10.7.5). The processor was a 2.93 Ghz Intel Core 2 DUO. The memory was 4 GB 1067 MHz DDR3. All results regarding performance should be interpreted as valid only for this (typical) machine configuration.

3. Results

3.1. Level of the reproducibility hierarchy reached

The system was available, as advertised. It installed without problems. The code did not run as advertised in the documentation, but the authors responded quickly to requests for help, and it was possible to get it working relatively quickly. Thus, level 3—executability—was reached.

3.2. Performance of pubmed.mineR package

The tester was unable to load any documents by following the documentation provided with the library. The authors responded quickly to requests for help, and the tester was successful in using one of the methods for loading data. Thus, level 3—executability—was reached for this library, as well.

3.2.1. Test 1. Performance of pubmed.mineR package on diverse document collections

In order to get a broad picture of pubmed.mineR performance, we evaluated it on four sets of data from PubMed. We varied the size of the data sets from quite small (about 2200 abstracts) to reasonably large (about 640,000 abstracts). It is not possible to keep the contents constant while varying size, so we tried instead to maximize variability of content by using four different queries to retrieve the abstracts. We then evaluated the mean processing time per document for each of the data sets.
Table 2 shows the queries for the four data sets.

Table 2: Queries and sizes for the four data sets.

Number of abstracts	query
2K (2,283 abstracts)	*synthetic lethal*
60K (59,854 abstracts)	*human drug disease "blood cell"*
172K (171,955 abstracts)	*human drug disease cell*
638K (637,836 abstracts)	*human drug disease*

Table 3: Per-document processing time varies with input size.

Data set	2K	60K	172K	638K
Size of file	4,148Kb	111,595Kb	153.5Mb	322.7Mb
Mean processing time per abstract (seconds)	0.0025	0.00025	0.000022	NA

The results are not problematic at all, but they are certainly unexpected. Processing time per document decreases quite a bit as document set size goes up. To evaluate the likely explanation that this was due to the length of the connection time to PubTator being amortized over a successively larger number of documents, we revisited the R code to ensure that we were only measuring the document processing time per se. Indeed, we found that the probable explanation was not the case at all, and that the unusual result does, in fact, represent the actual document processing times. We have no explanation for the behavior.

3.2.2. Test 2. Performance of pubmed.mineR package concerning its own stability

The second test evaluated the pattern of increase in processing time for the entire document collection, as well as variability in that processing time. The largest data set was used as input to pubmed.mineR, and the processing time was measured for every 10,000 abstracts. To evaluate variability, the process was repeated 10 times.
Figure 2 shows the cumulative processing time for the document collection and the mean processing time per document. The cumulative processing time for the document collection increases linearly. Figure 3 shows the variability in cumulative processing time over the course of the 10 repetitions. There are a few outliers, but the variation is generally small.

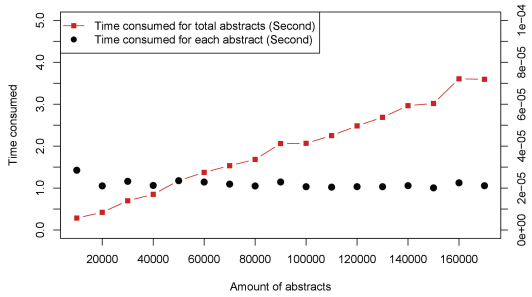


Figure 2: Cumulative processing time for the entire document collection and per-document processing time.

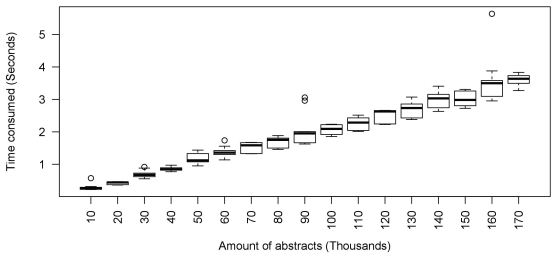


Figure 3: Variability in cumulative processing time.

3.3. Metamorphic testing and exploring the parameter space with rentrez

The functionality for pubmed.mineR and for rentrez are quite different. Rentrez's functionality is oriented less towards processing documents than towards retrieving them–more specifically, towards retrieving document identifiers. For an information retrieval library, different kinds of validation are applicable. In the case of rentrez, we tried metamorphic testing, and exploration of the parameter space.
Metamorphic testing (Murphy et al., 2008; Chen et al., 2009; Xie et al., 2009; Xie et al., 2011) is applied in situations where we have no "oracle"—situations where we

cannot know in advance what the exact output of a function or of a program should be. The general approach of metamorphic testing is to change some aspect of the input for which we can predict in a general way whether or not there should be a change in the output, and what the overall trend in the change should be. For example, if we calculate the mean and the standard deviation for some data set, and then add 100 to every value in the data set, the mean should change, and it should increase. In contrast, the standard deviation should not change at all.

The metamorphic test that we applied to rentrez was to give it two different queries, where we had a priori reason to think that the two queries should give us result sets of different sizes, and that the size of the second result set should be considerably smaller. For the first (and presumably larger) result set, we used the query *apoptosis*. (Apoptosis is a cellular process that is very important in embryological development and in the maintenance of proper cell states in the adult. Failure of apoptosis is a common cause of cancer, and due to the importance of apoptosis in development and in disease, it has been studied extensively (Hunter, 2012).) For the second (and presumably smaller) result set, we used the query *judo*. (Judo is a sport of Japanese origin that is not widely practiced in the United States, the source of most PubMed/MEDLINE publications, and it has not been widely studied in the English-language scientific literature.)

3.4. Rentrez methods: Exploring the parameter space

A search of PubMed/MEDLINE can return millions of article identifiers, but by default, the rentrez interface only returns 20 of them. This number can be changed by setting the appropriate variable when the search function is called. We varied the value of the variable systematically through a segment of the parameter space for that variable, which has no explicitly stated range, from 100000 to 1. We used the apoptosis query described in the methods for metamorphic testing.

3.5. Rentrez results: exploring the parameter space

We tried a realistic value for the variable that controls the maximum number of identifiers returned in a result set, as described above in the Methods section. We immediately found a bug. When the variable is set to 100,000 declaratively, the search function returns no identifiers. On the other hand, if it is set programmatically (e.g. in a for-loop from 100,000 down to 1), then the search function works well. After communication with the library author, a bug report has been filed.

3.6. New documentation

As one product of the study, new documentation was written for the two R libraries. It is available at https://zipfslaw.org/2015/10/19/pubmed-miner/ and at https://zipfslaw.org/2015/12/24/rentrez/, and has been accessed an average of once or more per day for the past several months. It is hoped that the documentation itself will add to the reproducibility of the work, by providing clear guidelines for running the code that were absent in the original publications—that is, by making it easier for future users to reach level 3 of the reproducibility hierarchy of needs.

4. Discussion and conclusions

4.1. Summary of the results of the two case studies

4.1.1. Pubmed.mineR

- The library is available and installs without problems. Problems with running the code were quickly resolved by communication with the authors, and new documentation addresses those issues. Level 3, or executability, was reached.
- Processing time for the document collection increases linearly with the size of the input data set.
- Processing time per individual document decreases with the size of the input data set.
- Processing time is relatively stable across multiple repetitions of the same input.
- New documentation responding to the problems with running the code may increase the chances for reproducibility in the future.

4.1.2. Rentrez

- The library is available and installs without problems. There were no problems with getting the code to run. Again, Level 3, or executability, was reached.
- The author was very responsive to requests for help with the library.
- Metamorphic testing did not reveal any issues.
- Exploring the parameter space quickly revealed an issue.

4.2. Conclusions

We have laid out some of the issues that pose problems for the notion of reproducibility in natural language processing. We showed how previous work on reproducibility in computer science can be used to establish a hierarchy of desiderata regarding reproducibility even in the face of these restrictions. We then showed how how those desiderata could be extended with conceptually straightforward, easily implementable tests of program performance.

The work reported here examined two R libraries for natural language processing in the biomedical domain. Both of those libraries presented the same problems for reproducibility testing: unavailability of data, and inadequate space for documentation of the experimental methodology in the original publications. We explored a number of possibilities for an expanded notion of reproducibility that we suggest might be appropriate for natural language processing research. In so doing, we found that all of those exploratory methods made a contribution to understanding the extent to which the research was or was not reproducible, whether by finding issues with the library (varying

input sizes until a library crashed; exploring the parameter space) or by providing reassuring sanity checks (metamorphic testing, stability). There is a possible counter-argument to the entire methodology of this paper. This counter-argument would be that papers in natural language processing describe *research,* not engineering efforts, and that therefore it is not relevant to study systems with respect to performance characteristics like processing times, the ability of the code to build, and the like. This counter-argument does not hold, because unlike the case in general natural language processing, the stated motivation in paper after paper in the biomedical natural language processing field is to provide a tool that meets some need of either physicians or biologists. The selection of biomedical natural language processing libraries for the work reported here was quite deliberate, as issues of run-time repeatability are quite relevant in this domain.

In principle, reproducibility in computational systems can be achieved easily without really addressing the right underlying issue. It should be possible to package arbitrary environments in a self-contained virtual machine that will execute for a long time to come. However, it may still not be possible to change anything about it or to use it for any actual task, and the fact that it produces the same output every time one pushes "run" is of little reassurance with respect to correctness, or even with respect to doing what is described in the paper. What one wants of a scientific result is to be able to (1) rely on it, and (2) put it to new uses (Fokkens et al., 2013). So, although the methodology described here improves on the lesser alternatives of not running, not building, or not even being available, evaluating the extent to which a program meets the goals of reliability and applicability to new uses remains for future work.

It is not our intention to point fingers or to lay blame at anyone's feet. As pointed out in the introduction to this paper, we are well aware that we ourselves have made software publicly available that no longer runs, or no longer functions completely, or is no longer available, despite our having published URLs for it. Rather, the hope of this paper is to help to establish a methodology for exploring issues of reproducibility in the field of natural language processing. It's clear that such a methodology is needed, and this paper does not claim to be the last word on the subject: two hours after we submitted this paper for review, one of the libraries stopped working completely—it returned only empty vectors. Perhaps the back end to which it connects had changed its interface—we really don't know, and the authors of the library have been silent on the specifics. After some communication with the authors, the former functionality was restored—the vectors that are returned are no longer empty. However, the behavior has not been the same since—that is, the contents of the vectors are different—and so far, we have been unable to reproduce our previous results. The research that relied on the library is at a standstill.

Acknowledgments

KBC's work was supported by grants NIH 2R01 LM008111-09A1 NIH 2R01 to Lawrence E. Hunter, LM009254-09 NIH to Lawrence E. Hunter, 1R01MH096906-01A1 to Tal Yarkoni, and NSF IIS-1207592 to Lawrence E. Hunter and Barbara Grimpe.

5. Bibliographical References

Barrows, N. J., Le Sommer, C., Garcia-Blanco, M. A., and Pearson, J. L. (2010). Factors affecting reproducibility between genome-scale sirna-based screens. *Journal of biomolecular screening*, 15(7):735–747.

Begley, C. G. and Ellis, L. M. (2012). Drug development: Raise standards for preclinical cancer research. *Nature*, 483(7391):531–533.

Berez, A. L. (2015). On valuing reproducibility in science and linguistics. *Research, Records and Responsibility: Ten years of PARADISEC*, page 39.

Bisang, W. (2011). Variation and reproducibility in linguistics. *Linguistic Universals and Language Variation*, 231:237.

Branco, A. (2012). Reliability and meta-reliability of language resources: Ready to initiate the integrity debate? In *Proceedings of the twelfth workshop on treebanks and linguistic theories*, pages 27–36.

Chen, T. Y., Ho, J. W., Liu, H., and Xie, X. (2009). An innovative approach for testing bioinformatics programs using metamorphic testing. *BMC Bioinformatics*, 10(1):24.

Collaboration, O. S. et al. (2012). An open, large-scale, collaborative effort to estimate the reproducibility of psychological science. *Perspectives on Psychological Science*, 7(6):657–660.

Collberg, C., Proebsting, T., and Warren, A. (2014a). Repeatability and benefaction in computer systems research: A study and a modest proposal. *Department of Computer Science, University of Arizona, Tech. Rep. TR*, pages 14–04.

Collberg, C., Proebsting, T., Moraila, G., Shankaran, A., Shi, Z., and Warren, A. M. (2014b). Measuring reproducibility in computer systems research. *Department of Computer Science, University of Arizona, Tech. Rep*.

Collins, F. S. and Tabak, L. A. (2014). Nih plans to enhance reproducibility. *Nature*, 505(7485):612.

Fokkens, A., Van Erp, M., Postma, M., Pedersen, T., Vossen, P., and Freire, N. (2013). Offspring from reproduction problems: What replication failure teaches us. In *Association for Computational Linguistics*, pages 1691–1701.

Hirschman, L., Bourne, P., Cohen, K. B., and Yu, H. (2007). Translating Biology: text mining tools that work.

Hunter, L. E. (2012). *The Processes of Life: An Introduction to Molecular Biology*. The MIT Press.

Ihaka, R. and Gentleman, R. (1996). R: a language for data analysis and graphics. *Journal of computational and graphical statistics*, 5(3):299–314.

Kovacevic, J. (2007). How to encourage and publish reproducible research. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–1273. IEEE.

Mobley, A., Linder, S. K., Braeuer, R., Ellis, L. M., and Zwelling, L. (2013). A survey on data reproducibility in cancer research provides insights into our limited abil-

ity to translate findings from the laboratory to the clinic. *PLoS One*, 8(5):e63221.

Murphy, C., Kaiser, G. E., and Hu, L. (2008). Properties of machine learning applications for use in metamorphic testing.

Pedersen, T. (2008). Empiricism is not a matter of faith. *Computational Linguistics*, 34(3):465–470.

Poprat, M., Beisswanger, E., and Hahn, U. (2008). Building a BioWordNet using WordNet data structures and WordNet's software infrastructure–a failure story. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 31–39, Columbus, Ohio, June. Association for Computational Linguistics.

Prinz, F., Schlange, T., and Asadullah, K. (2011). Believe it or not: how much can we rely on published data on potential drug targets? *Nature reviews Drug discovery*, 10(9):712–712.

Proebsting, T. and Warren, A. M. (2015). Repeatability and benefaction in computer systems research.

Rani, J., Shah, A. R., and Ramachandran, S. (2015). pubmed.mineR: An R package with text-mining algorithms to analyse PubMed abstracts. *Journal of biosciences*, 40(4):671–682.

Vandewalle, P., Kovačević, J., and Vetterli, M. (2009). Reproducible research in signal processing. *Signal Processing Magazine, IEEE*, 26(3):37–47.

Wei, C.-H., Kao, H.-Y., and Lu, Z. (2013). PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Research*, page gkt441.

Xie, X., Ho, J., Murphy, C., Kaiser, G., Xu, B., and Chen, T. Y. (2009). Application of metamorphic testing to supervised classifiers. In *Quality Software, 2009. QSIC'09. 9th International Conference on*, pages 135–144. IEEE.

Xie, X., Ho, J. W., Murphy, C., Kaiser, G., Xu, B., and Chen, T. Y. (2011). Testing and validating machine learning classifiers by metamorphic testing. *Journal of Systems and Software*, 84(4):544–558.

Gonito.net – open platform for research competition, cooperation and reproducibility

Filip Graliński*, Rafał Jaworski*, Łukasz Borchmann†, Piotr Wierzchoń†

Adam Mickiewicz University
*Faculty of Mathematics and Computer Science
†Institute of Linguistics
*ul. Umultowska 87, 61-614 Poznań, Poland
†al. Niepodległości 4, 61-874 Poznań, Poland
{filipg,rjawor,wierzch}@amu.edu.pl, borchmann@rainfox.org

Abstract

This paper presents the idea of applying an open source, web-based platform – Gonito.net – for hosting challenges for researchers in the field of natural language processing. Researchers are encouraged to compete in well-defined tasks by developing tools and running them on provided test data. The researcher who submits the best results becomes the winner of the challenge. Apart from the competition, Gonito.net also enables collaboration among researchers by means of source code sharing mechanisms. Gonito.net itself is fully open source, i.e. its source is available for download and compilation, as well as a running instance of the system, available at `gonito.net`. The key design feature of Gonito.net is using Git for managing solutions of problems submitted by competitors. This allows for research transparency and reproducibility.

1. Introduction

The field of Natural Language Processing struggles with numerous problems regarding research reproducibility and reuse. A common practice for the researchers is to focus on a very specific problem and engineer software that tackles it. Continuation of research is understood as further engineering of the same software for the same basic problem, with minor changes to the requirements. Even if such a project is carried out by a group of researchers rather than an individual, it should still be viewed as isolated. At the end of the project, the software might be released as an autonomous tool (open source or closed). However, even if an NLP tool gets released, it is often seen as a black box for other researchers, who use it as a submodule for isolated projects of their own.

As a result, many NLP researchers invent solutions they claim are specific for their project, while in reality these solutions could easily be generalised. Furthermore, when working on their projects, researchers tend to "reinvent the wheel" (even when working on the same data set). Instead of taking advantage of someone else's work they start from scratch and try to solve previously solved problems.

One effort to break the effect of isolation of NLP projects is organising shared tasks. In this scenario numerous researchers work on one specific, very well defined problem. Their work is organised as a competition, which is a highly motivating factor. However, the participants' work is still isolated, as they share experiences only after the shared task results are published.

This article focuses on a modified version of the shared tasks approach. It consists in organising shared tasks/challenges where the results and solutions submitted by participants could be open and ready to inspire other participants. The whole process is administered by the *Gonito.net* platform – custom made, open source software.

Similar platforms: Kaggle, CodaLab and others are described in Section 2. Section 3. lists related work examples. The Gonito.net platform itself is presented in detail in Section 4. Section 5. is a Gonito.net walkthrough. Lastly, ideas for improving citation of resources and tools and final conclusions are formulated in Sections 6. and 7.

2. Similar platforms

2.1. Kaggle

Kaggle is a commercial platform for organising competitions in the area of data science and corpus research, including natural language processing. It is a meeting point for business enterprises and data scientists, who are willing to tackle specific problems suggested by the enterprises.

In a typical Kaggle usage scenario, a large corporation proposes a problem, whose solution would be beneficial for its cause. In order to solve the problem, a significant amount of data has to be processed, using, potentially, a wide variety of methods of data analysis. The corporation provides the data and offers a tempting monetary prize (usually between 10 000 – 100 000 US dollars) for the scientist who is able to achieve the best analysis results. For instance, one of the active challenges in February 2016 consisted in predicting the relevance of search results on the *homedepot.com* website and offered a 40 000 US dollars prize. There are also challenges with no monetary prizes at all.

However, while submitting problem solutions is free of charge, proposing the problem (referred to as "hosting a challenge") is not. For that reason, subjects who host the challenges are referred to as "sponsors". Limitations in hosting challenges (with monetary prizes or without) are to ensure that only important, real-life problems are solved at Kaggle. Nevertheless, Kaggle also has a separate module – "Kaggle in class" – designed specifically for academic institutions. In this module, a university teacher can host a challenge for his or her students, without having to pay for that service (with some formal limitations).

As Kaggle deals with real problems and, more importantly, with real money (often considerable amounts), the platform implements a carefully designed system of scoring the users in the challenges and generating leaderboards. This is done mainly to ensure the fairness of the competition. Firstly, there is a distinction between *public* and *private* leaderboards. Users' submissions are first evaluated on development sets. These results are published in the public leaderboards. The submissions are evaluated on test sets, which are not made available for the users during their work. The results achieved on the test sets are used to generate the private leaderboards. These private leaderboards are made public only after the end of the challenge and serve to determine the final ranking. This is to prevent the users from overfitting their models to the test data.

To summarise, Kaggle incorporates many valuable concepts, such as the idea of competition between users or the system of scoring and generating leaderboards. However, the main drawback of Kaggle is the fact that it is a closed, commercial system and the cost of hosting challenges is often too high for individual researchers.

2.2. CodaLab

CodaLab (available at `codalab.org`) is a platform for hosting machine learning challenges, following similar principles as Kaggle. The main difference, though, lies in the fact that at CodaLab both entering and hosting a challenge are free of charge. Furthermore, the whole platform is open.

From a competitor's point of view, work on a challenge consists of:

1. downloading data for analysis from CodaLab in a so-called "competition bundle",
2. developing analysis tools,
3. wrapping results in a "reference bundle",
4. uploading the bundle to CodaLab.

Bundle is CodaLab's format for data interchange. Technically, it is a zip archive of a specific file structure. An example competition bundle is shown in Figure 1: `reference.zip` is the bundle storing the results, whereas `program.zip` is a bundle holding the sources of the program that generated the results.

Figure 1: CodaLab's competition bundle

```
competition.zip
  |- competition.yaml
  |- data.html
  |- evaluation.html
  |- logo.jpg
  |- overview.html
  |- program.zip
  |- reference.zip
  |- terms_and_conditions.html
```

While the openness of the platform is a significant advantage, the use of bundles for data interchange might not be comfortable. It would require a separate tool just to handle the bundle packing and unpacking process. Furthermore, CodaLab does not use any version control system to track the changes in submitted programs.

2.3. DrivenData

There is a competition hosting platform very similar to Kaggle, called DrivenData (`drivendata.org`). It is also a closed system, but it enables non-profit organisations to host challenges. Still, the platform is not well suited for academic purposes, as the challenges must bring solutions to practical problems of the hosting organisations.

2.4. Numerai

Numerai is a service available at `numer.ai`, hosting a worldwide tournament for machine learning special-

ists. The sole task in the tournament is predicting the stock market. The competition is possible thanks to the fact that Numerai provides anonymised stock market data and makes it publicly available. Top users are rewarded with money prizes.
It is fair to say that Numerai succeeded in organising a world-wide competition for data scientists. However, Numerai is not applicable to scientific research, as the platform is completely closed and cannot be used for any tasks other than stock market prediction.

3. Related work examples

The problem of research reproducibility has already been addressed in various ways. Some of the existing solutions to the problem use Git as the core system for managing development workflow. This section presents some of the most interesting setups, ensuring full reproducibility of the results, tidiness of resource storage and the transparency of the research.

3.1. Git-based setup

(Ram, 2013) describes a research workflow managed by the Git version control system. The idea of applying Git for this task is inspired by well-known software engineering findings, stating that version control systems (VCS) are crucial for managing resources especially in environments with multiple developers (see (Spinellis, 2005)). The primary functionality of a VC system is tracking changes in text files. Each change made by the developers in a file is bound with an informative comment, thus providing Git with exhaustive file history data. Not only does the history contain all text changes, but the reason for each change can be inferred from the comments.
Another distinctive feature of Git are branches. While all changes of the file are typically tracked in one history timeline (called the `master` branch), it is possible to create multiple branches. When a new branch is created, all changes to the file can be tracked either in the master branch, or the newly created branch. At any time, all changes from the new branch can be merged into the main branch. The branching mechanism is particularly useful when a developer needs to modify the file in a way that might make the file unusable for collaborators. In this case, the developer makes all the necessary modifications in his or her own branch and only after the changes are thoroughly tested, are they merged with the main branch.
Here are example use cases of the Git setup described in (Ram, 2013):

- creating lab notebooks, edited by multiple users,
- facilitating collaboration with the use of branches,
- backup and failsafe against data loss,
- freedom to explore new ideas and methods,
- increased transparency and verifiability.

3.2. Git and org-mode based setup

(Stanisic et al., 2015) describe more advanced ideas on preparing environment for reproducible research. By taking advantage of the Git branching feature, the authors developed their own branching model and defined typical operations on it. As Git is best suited for tracking changes in text files, the authors enhanced the environment with the use of the org-mode software, enabling task managing and organising work by commands written in plain-text format.
With the use of this environment, the authors managed to publish a fully reproducible paper on parallel computing: (Stanisic et al., 2014). However, although the Git repository itself is made publicly available, it is not ready for direct introduction to other research workflows.

4. The Gonito.net platform

The Gonito.net web application is an open-source platform for machine learning competitions. The idea is simple, yet powerful and malleable:

- challenges are published on the platform (at least as test sets, of course training and development sets could be provided as well),
- users can submit their solutions (the system output for the test set, possibly accompanied with the source codes or even full papers describing the system),
- the results are automatically evaluated by the system.

The results can then be tracked in terms of:

- timeline (who submitted what and when?),
- performance (which solution is the best at the moment? i.e. a leaderboard is presented),
- provenance (what is based on what? what was forked from what?).

The Gonito.net system is founded on two key (and interdependent) principles:

Be open
- Gonito.net is available[1] as open-source software under GNU Affero General Public License.

[1] `git://gonito.net/gonito` or `http://gonito.net/gitlist/gonito.git`

- Anyone can set up their own instance of Gonito.net (whether local or not) and run whatever challenges they please.
- Users are encouraged (but not forced) to share the source codes of their solutions.
- Users are free to use whatever programming language and tools to generate the output (only Git, a widely adopted tool, is required).

Use Git

- Solutions can be uploaded to the Gonito.net platform with Git without clicking around and uploading files in a browser.
- New challenges are created as Git repositories.
- With Git it is possible to track which solution was forked and reused in another solution.
- Even if a Gonito.net platform ceases to exist, the results and source codes may still be available as a regular Git repository to be cloned and inspected with standard Git tools, no external database is needed.

The first principle differentiates Gonito.net from Kaggle, whereas the second one – from CodaLab.
Notwithstanding the simplicity of the idea, Gonito.net can support a variety of workflows and could be used for a wide range of different purposes:

- as an auxiliary teaching tool (for machine learning or NLP classes) helping to keep track of students' assignments and progress (just as in "Kaggle in class", but students' identity and work do not have to be shared with any external company – if privacy is a concern),
- within a company when working on a machine learning problem,
- within a small group of researchers to keep track of the progress (e.g. when working on a paper),
- for organising shared tasks,
- for tracking effort of a given research community on standard tasks in a longer-term perspective (not just as a one-off event).

Actually, a Gonito.net challenge can go through a combination or sequence of such stages: it can be started and "test-run" as a student course assignment, then a small group of researchers could work on it locally and after some time the challenge could be made public as a shared task competition, but even when the shared task is completed, the Gonito.net platform could be used to continuously track the progress of the whole research community on a given problem. If adopted, Gonito.net could provide the focal point where the best solution for a particular problem might become common knowledge ("everyone knows that everyone knows... that the current best result is there").
A Gonito.net challenge does not have to be only about the competition – an appropriate blend of competition and cooperation could be achieved with Gonito.net, as Git makes it easy to reuse other peoples' solutions and build on them. (Just as science, in general, is a mixture of "racing to the top" and "standing on the shoulders of giants".)
Gonito.net is written in Haskell (using the Yesod web framework) and is accompanied with *GEval*, a library and stand-alone tool for machine learning evaluation. At this time, the accuracy, (root-)mean-square error and BLEU metrics are implemented in GEval. Due to the application author's background in natural language processing the stress has been on NLP metrics and challenges so far. Nevertheless, Gonito.net could be used for any machine learning challenge, provided that an evaluation metric is implemented in GEval.
An instance is available at `http://gonito.net` along with some sample (but non-toy) NLP challenges (of course, anybody can set up another instance, whether it is to be publicly or only internally available). For example, one of challenges is about guessing the publication year (1814-2013) of a short Polish text.[2] Another challenge will be described in Section 5..
The sample instance is accompanied with a Git-hosting system (Gitolite + GitList). The whole assembly is packaged and made available as a virtual machine image so that anybody could get it up and running quickly if it is to be self-hosted.

5. Gonito.net walkthrough

This section presents a walkthrough of participating in a challenge from a competitor's point of view. The challenge used in this walkthrough will be one of Gonito.net currently running challenges: the "He Said She Said Classification Challenge". The task is to identify, whether a Polish text is written by a male or female.
The corpus used in this task was obtained by processing the Common Crawl-based Web corpus of Polish (Buck et al., 2014), using the method described in detail in (Graliński et al., 2016). Author's gender

[2] http://gonito.net/challenge/retroc

determination was possible thanks to the existence in Polish of gender-specific first-person expressions (for instance, *I said* will be rendered in Polish as *powiedziałem* or *powiedziałam* depending on whether it was spoken or written by a man or a woman). Text fragments without such expressions were discarded. Then, a gender-neutral version of the corpus was prepared and made available on Gonito.net for the purposes of training and testing classifiers.

5.1. Prerequisites

Upon starting a challenge it is assumed that the competitor is familiar with Git and a Unix-like operating system environment.

The first step[3] of setting up Gonito.net's specific environment is installing the GEval software. It is written in the Haskell programming language and requires the Haskell Stack program, available for download at `https://github.com/commercialhaskell/stack`. With Haskell Stack installed, the following commands install GEval:

```
git clone git://gonito.net/geval
cd geval
stack setup
stack install
```

The simplest way to get the challenge data is to clone the read-only repository, e.g. for the "He Said She Said" challenge the URL is `git://gonito.net/petite-difference-challenge` (also reachable and browsable at `http://gonito.net/gitlist/petite-difference-challenge.git/`). Then a user can use his or her own repository, wherever it is hosted (at his or her private server, at GitHub, etc.). In sections 5.2.–5.8., however, a slightly more complicated (but recommended in the long term) workflow using repositories hosted at Gonito.net is presented.

5.2. Getting the work repository

Every user registered at Gonito.net receives their own Git repository for each challenge they might participate in. In order to start working, the user must first enter his or her login name and a SSH public key at `http://gonito.net/account` and then clone the repository. Assuming the user login name is `tom` and challenge ID: `petite-difference-challenge` (actual ID of the "He Said She Said" challenge), the following commands are used to prepare the user's repository:

```
git clone ssh://gitolite@gonito.net/
  tom/petite-difference-challenge
cd petite-difference-challenge
git pull ssh://gitolite@gonito.net/
  petite-difference-challenge
git push origin master
```

Note that the first command clones an empty repository – the private repository of the user `tom`. The third command is used to pull the contents of the mother repository, containing the challenge data and solution template.

The solution template contains the following files and directories:

- `README.md` – text file with a detailed challenge description,
- `train` – folder with training data, usually compressed plain text file in tab separated values (TSV) format,
- `dev-0` – folder with annotated corpus, containing input and expected data (files `in.tsv` and `expected.tsv` respectively),
- `test-A` – folder with test corpus, containing only input data.

(More development and test sets may be added later by the challenge organisers.)

5.3. Working on solution

While working on a solution, a user is required to develop a data analysis tool able to produce predictions for data in the `dev-0/in.tsv` and `test-A/in.tsv` files. These predictions should be stored in the `dev-0/out.tsv` and `test-A/out.tsv` files respectively. With the help of the GEval software, a user is able to evaluate his or her results on the development set locally, before submitting them to Gonito.net. In order to do this, the following command[4] must be issued from within the top directory of the solution repository:

```
geval --test-name dev-0
```

[3] Actually, this step is not obligatory, as a user could just use the web application to evaluate his or her submissions. On the other hand, it is recommended to check the solution locally on the development set before submitting to the web application in order to make sure it is correctly prepared and formatted.

[4] It is assumed that `~/.local/bin` is added to the `$PATH` environment variable.

In the "He Said She Said" challenge, the result is one number representing the accuracy of guesses.
When a user is satisfied with the performance of the software on the development set, he or she may try to submit the solution to Gonito.net and check the results on the test set.

5.4. Submitting a solution

Submitting a solution to Gonito.net consists of two steps – sending output files via Git and notifying Gonito.net of the changes in the repository.
In order to send the output files via Git one should issue the following commands:

```
git add dev-0/out.tsv test-A/out.tsv
git commit -m 'my brilliant solution'
git push origin master
```

In the second step, the user must click the "Submit" button[5] in Gonito.net and provide the following information:

- informative submission description,
- repository URL (auto-filled to the URL of the user's repository on Gonito.net),
- repository branch (auto-filled to `master`).

Alternatively, instead of manually clicking the "Submit" button, a Git "hook" can be configured to get Git to notify Gonito.net of a commit and to trigger an evaluation automatically (just as continuous integration servers are often configured to be triggered whenever a new commit is pushed).
All submissions for a challenge are evaluated on both the development and test set. All these results are visible to all participating users (the user names could be anonymised if desired).

5.5. Best practices

Although it is not formally required, the users are encouraged to submit the sources of their programs along with the results via their Git repositories. A good idea is to provide a Makefile which could facilitate the process of running the program on the data. However, the recommendation for submitting the sources of the program excludes statistical models and any other resource files that can be generated by the software. (The authors of Gonito.net are planning incorporating git-annex Git extension into the Gonito.net setup, so that models and other large files could be uploaded and downloaded there, cf. (Korolev and Joshi, 2014)). Furthermore, if the tool uses randomisation at any step of the analysis, this randomisation should use a specified seed number.
All these recommendations help to ensure transparency and reproducibility of the solutions.

5.6. Example solutions to the "He Said She Said" challenge

Let us now prepare a simple, baseline solution for the "He Said She Said" problem. The solution will always guess that the text is written by a male. Sample code for this solution can be given as a shell script:

```
#!/bin/sh

filename=$1/in.tsv

while read -r line
do
    echo M >> $1/out.tsv
done < "$filename"
```

Let us also prepare a Makefile:

```
all: classify-dev classify-test geval

classify-dev: dev-0/out.tsv

classify-test: test-A/out.tsv

dev-0/out.tsv:
  ./classify.sh dev-0

test-A/out.tsv:
  ./classify.sh test-A

clean:
  rm -f dev-0/out.tsv test-A/out.tsv
```

Now, after issuing the commands `make clean` and `make` we obtain the files `dev-0/out.tsv` and `test-A/out.tsv`. Checking the results achieved on development set (`geval --test-name dev-0`) gives the expected result of 0.5 (as the test corpora are evenly balanced in terms of the number of male and female texts).

5.7. Availability and reproducibility of a submission

This submission described in Section 5.6. was made public as commit 86dd91 and, consequently:

- in the electronic edition of this paper, the above commit number prefix

[5] See `http://gonito.net/challenge-submission/petite-difference-challenge`

is clickable[6] as `http://gonito.net/q/86dd914ad99a4dd77ba1998bb9b6f77a6b076352` and leads to a submission summary,

- the submission (both the output data and source codes) is available as `git://gonito.net/petite-difference-challenge` (branch `submission-00072`) – anybody can clone the repository, inspect the data and source codes, try to reproduce the results (with `make clean && make`) and create their own solution based on them,
- the submission is accessible with a Web browser at `http://gonito.net/gitlist/petite-difference-challenge.git/submission-00072/`,
- as Git commit ID (SHA-1 hash) is used to identify a submission, the submission data might be available even if Gonito.net stops working (provided that it was pushed to some external repository, e.g. GitHub).

5.8. Non-trivial submission

A less trivial approach can be implemented in Python with scikit-learn machine learning library (Pedregosa et al., 2011), and be designed as follows:

1. convert text to a matrix of token counts, previously lower-casing it, and stripping all the punctuation,
2. make term-document matrix from joined punctuation marks' n-grams,
3. concatenate results of the above transformer objects with `FeatureUnion`,
4. transform the above count matrix to a normalised TF-IDF representation,
5. train logistic regression classifier.

Such a solution received an accuracy score above 0.63 and is available with the source code on the Gonito.net (commit 2cb823{model.py}).
The leaderboard with the graph presenting the submission times (x-axis) and the scores (y-axis) is given in Figure 2. Each point represents one submission and the relations between submissions (what was forked from what?) are represented with arrows.

[6]Alternatively, the ID `86dd91` could be entered manually at `http://gonito.net/q` or directly pasted into a link (`http://gonito.net/q/86dd91`)

6. Improving citation of tools and resources

A very simple, technical measure within the Gonito.net platform to improve citation of tools and resources is to introduce a standard location for a file (`references.bib`) where BibTeX references to the paper describing the solution (and the solutions forked and re-used so far) are kept. The file should be initiated with the reference to the paper describing the challenge itself and the related language resource. With each solution to be described in some paper, the `references.bib` file should be extended with a new entry.
This approach, though, has a weakness: the BibTeX entry is, obviously, available only when the paper is published, which is usually much later than the submission to Gonito.net is done. Actually, it is a more general problem: the final paper itself will usually be completed *after* the submission (even if it is written along with the software solution and some draft version is ready then). The workflow we propose is to push a new commit with the final paper and another one with the BibTeX entry, when they are ready and submit them to Gonito.net (with *the same* output data as the initial submission). The idea is that Gonito.net stores an extra check-sum of output files (just output files, excluding source codes) and the commits at various stages (but related to the same results) can be tracked by Gonito.net and presented together.

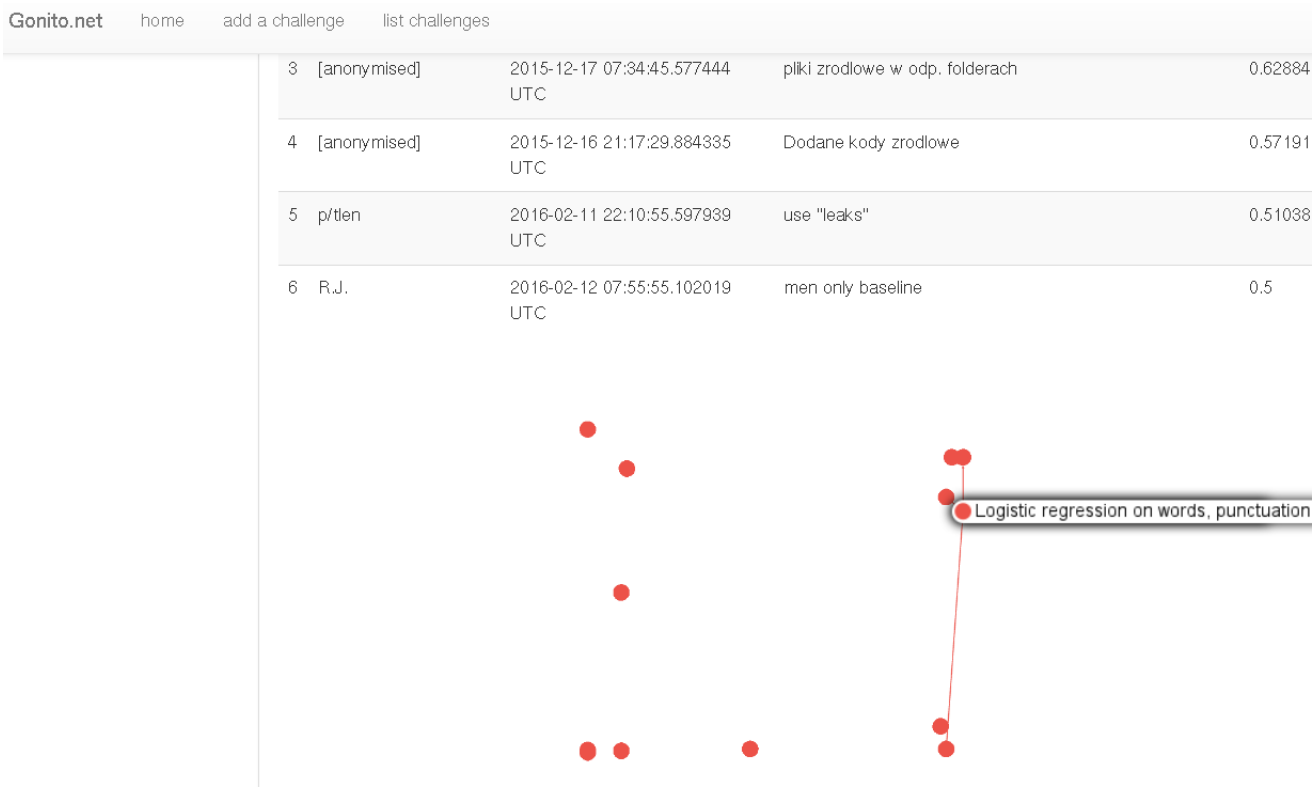
Acknowledgements

Work supported by the **Polish Ministry of Science and Higher Education** under the **National Programme for Development of the Humanities**, grant 0286/NPRH4/H1a/83/2015: "50 000 słów. Indeks tematyczno-chronologizacyjny 1918-1939".

7. Conclusions and future work

Gonito.net is currently hosting three challenges, being tackled by a small community of researchers and students at an academic institution. With this article we suggest that other researchers try to apply Gonito.net for hosting NLP or other types of scientific tasks. Importantly, the openness of Gonito.net makes it applicable in any research workflow. After the first months of using the platform it is possible to conclude that Git proves extremely useful in managing the process of storing and exchanging of tools and resources.
Ideas for future work include further increasing the transparency of submitted solutions by providing mechanisms facilitating code reuse. Ideally, a new solution for a problem could be based on the current best solution. The researcher should try to improve the best

Figure 2: Gonito leaderboard with a graph representing the submissions and the relations between them



solution and bring the score to a higher level. Thus, the competition could become a productive cooperation.

8. References

Buck, Christian, Kenneth Heafield, and Bas van Ooyen. 2014. N-gram counts and language models from the common crawl. In *Proceedings of the Language Resources and Evaluation Conference*, Reykjavk, Icelandik, Iceland, May.

Graliński, Filip, Łukasz Borchmann, and Piotr Wierzchoń. 2016. He said she said – male/female corpus of polish. In press, to appear in Proceedings of the LREC 2016 Conference.

Korolev, Vlad and Anupam Joshi. 2014. Prob: A tool for tracking provenance and reproducibility of big data experiments. *Reproduce'14. HPCA 2014*, 11:264–286.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ram, Karthik. 2013. Git can facilitate greater reproducibility and increased transparency in science. *Source Code for Biology and Medicine*, 8(1):1–8.

Spinellis, Diomidis. 2005. Version control systems. *Software, IEEE*, 22(5):108–109.

Stanisic, Luka, Samuel Thibault, Arnaud Legrand, Brice Videau, and Jean-François Méhaut. 2014. Modeling and Simulation of a Dynamic Task-Based Runtime System for Heterogeneous Multi-Core Architectures. In *Euro-par - 20th International Conference on Parallel Processing*, Euro-Par 2014, LNCS 8632, Porto, Portugal, August. Springer International Publishing Switzerland.

Stanisic, Luka, Arnaud Legrand, and Vincent Danjean. 2015. An effective git and org-mode based workflow for reproducible research. *SIGOPS Oper. Syst. Rev.*, 49(1):61–70, January.

SHORTREF.ORG - Making URLs Easy-to-Cite

Jozef Mišutka, Ondřej Košarko, Amir Kamran
Institute of Formal and Applied Linguistics, Charles University in Prague
Malostranské Nám. 25, Prague
misutka@ufal.mff.cuni.cz, kosarko@ufal.mff.cuni.cz, kamran@ufal.mff.cuni.cz

Abstract

In this paper, we present an easy-to-cite and persistent infrastructure (`shortref.org`) for research and data citation in the form of a URL shortener service. The reproducibility of results is very important for the reuse of research and directly depends on the availability of research data. The advancements in web technologies made the redistribution of data much easier; however, due to the dynamic nature of the internet, the content is constantly on the move from one destination to another. The URLs researchers use for citing their work do not directly account for changes and when the users try to access the cited URLs, the URLs do not need to be working anymore. In our proposed solution, the shortened URLs are not basic URLs but use persistent identifiers and provide a reliable mechanism to make the target always accessible which can directly improve the impact of research.

Keywords: Repository, Citations, Persistent URL Shortener

1. Introduction

Much of today's research depends on the underlying data. The availability of data is very important in the reproducibility of results. However, a number of times, after the research findings are published the data is either lost or becomes dubious due to the lack of reliable citations of the data. Piwowar et al. (2007) showed that the impact and reuse of research is directly related to the public availability of the related data.

There are various initiatives e.g., Research Data Alliance[1] or FORCE11[2], supporting proper data citations. Together with data, online tools are also being developed to visualise and to query the data. These tools are often deployed in large infrastructures as web services, and can process large quantities of data. This makes the use of these tools easier without the need to install them or without having the computational resources. The data can be inspected and analysed online, but what is still missing is a way to uniquely identify and properly cite these dynamic results and subsets of the data. Even if the service has a well-defined API, the queries become too long to use them in a publication. One possible way to share the data analysis is by converting them into graphical content; however, to match a predefined publication format some of the important details are lost or hidden from view. A picture is also not easily verifiable.

What if the researchers could easily and reliably share and cite dynamic content (e.g., results of queries, visualisation of datasets, interactive graphs) and let the readers inspect the exact same analysis and subset of data? This will help in the reproduction of the results and the verification of the research findings and will also encourage data citations.

In this paper, a URL shortener (shortref.org) is described which uses the handle system (Kahn and Wilensky, 2006) to create persistent identifiers (PIDs[3]) and easy-to-cite URLs.

There are a number of URL shortener services (goo.gl, TinyURL, bit.ly etc.) already available and heavily used to share contents over the web, especially popular on social sharing platforms. The research community is also starting to adapt these services to cite digital resources. However, these shortener services only link the shortened URL to the target resource. When a resource is moved or becomes unavailable, these services do not keep track of the changes and return users with a 404 or not found error message. On the other hand, our proposed service architecture, with the help of linked metadata, can present useful information about the resource even if the resource is not available. The users can still contact the owner/maintainer of the resource and obtain the required data. This fallback procedure can be executed by adding a "noredirect" parameter to the end of the shortened URL e.g., http://hdl.handle.net/11346/PMLTQ-9XCM?noredirect. This will access the records available directly on the Handle server and present the page as shown in Figure 1.

Persistence is the key feature that distinguishes our service from other URL shortener services. Most available services claim that the generated URLs will never expire; however, these free service providers can at anytime decide to discontinue their service, which means that the cited links generated by these services can stop working. With shortref.org this is not the case, by using handles we managed to separate the generated links from our service. The handles are registered and maintained by the reliable Handle.net registry run by CNRI[4] and authorized by the DONA Foundation[5]. A number of projects and institutions are currently making use of the Handle System, including but not limited to: the Defense Virtual Library (DTIC), the DARPA and CNRI; the Digital Object Identifier System (DOI); the

[1] The Research Data Alliance Web Page. http://rd-alliance.org

[2] FORCE11 is a community of scholars, librarians, archivists, publishers and research funders. https://www.force11.org

[3] A persistent identifier is a long-lasting reference to a digital object.

[4] Corporation for National Research Initiatives. http://www.cnri.reston.va.us

[5] The DONA Foundation is a technical management organization that has taken over responsibility for the evolution of CNRI's Digital Object (DO) Architecture including outreach around the world.

Handle.Net®

Handle Values for: 11346/PMLTQ-RXYK

Index	Type	Timestamp	Data
100	**URL**	2015-10-27 09:24:30Z	https://lindat.mff.cuni.cz/services/pmltq/#!/treebank/ud_da/query/lYWgdg9gJgpgBAbQFBzsAZgVzHAvHAcg
11800	**TITLE**	2015-10-27 09:24:30Z	Universal Dependecies - Danish | Query | PML Tree-Query Engine
11801	**REPOSITORY**	2015-10-27 09:24:30Z	LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague
11802	**SUBMITDATE**	2015-10-27 09:24:30Z	2015-10-27T09:24:30Z
11803	**REPORTEMAIL**	2015-10-27 09:24:30Z	lindat-help@ufal.mff.cuni.cz

Handle Proxy Server Documentation
Handle.net Web Site

Figure 1: Fallback metadata recovery page, if the target resource is unavailable.

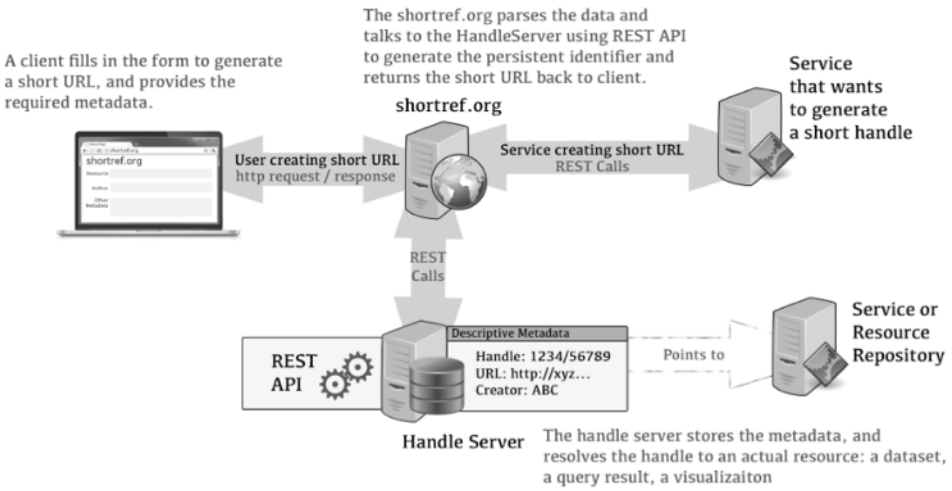


Figure 2: Workflow to generate a short persistent identifier.

digital repository software DSpace, which can use Handles to name and provide access to document containers.

2. Requirements

Citing dynamic results (e.g., citing interesting queries on actual datasets) should be similar to citing a dataset itself. The link used in a citation should be persistent. It should have descriptive metadata attached to describe and identify the contents. The persistent link should be reasonably short and readable. The metadata accompanying the link should be well defined and extensible. The provenance or origin of the record is required - title or description (or both), time of creation (stored automatically) and change history. Furthermore, contact information must be provided in case the final URL stops working.

The services and datasets that users are referencing will not be hosted at shortref.org. The main purpose is to provide a way to keep the links persistent during the lifetime of a service. The APIs can change and services can move from domain to domain. With the help of descriptive metadata it should be possible to identify the original resource in time and the creators of the resource. It is also possible to monitor the target URLs and in case of resolution failures users will be redirected to shortref.org showing the metadata associated with the resource. The information can be used to identify the query, service and data in time also containing the contact information for the responsible PID creator and/or service maintainer.

The service exposes REST API allowing for other services to simplify and automate the creation of shortref.org PIDs. The interaction can be hidden behind the graphical interface of the particular service.

The final requirement is identity management namely authentication and authorisation. This is needed in order to support updates to the PIDs from the web interface and through REST API.

3. Technical implementation

Handles are used as persistent identifiers (PIDs). A dedicated prefix is assigned; therefore, it can be completely moved out of the infrastructure if the need arises. Handles are not only mapping a PID to a URL, they are mapping a PID to an object (a key-value pairs) and we can make the metadata part of this object.

The default handle server provided by Handle.net is used. It is a JAVA application, which after registering your prefix becomes a part of the global handle system. Users send their queries to the global resolver to resolve PIDs belonging under your registered prefix. The global handle server then contacts the locally installed application registered un-

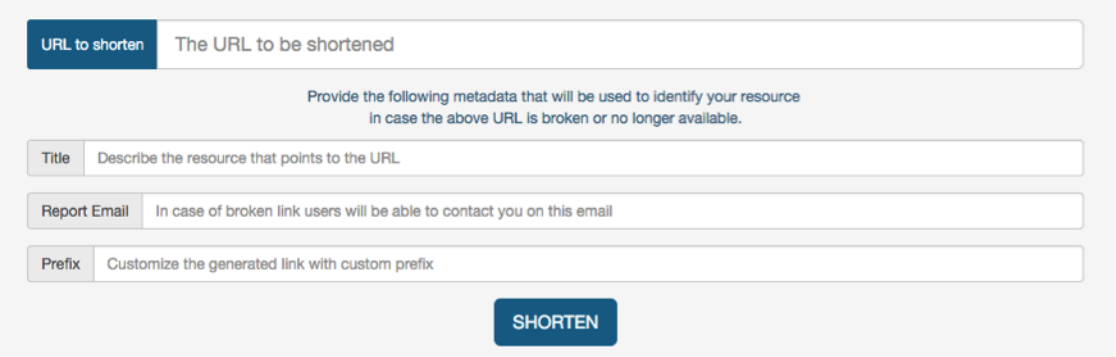


Figure 3: Web form to fill metadata and generate shortened URL.

der the prefix. A custom implementation of JAVA HandleStorage interface is provided that adds the metadata to the responses. For the data storage a SQL database is used. The resolution is a table lookup, where the handle is the unique key.

For the users, a simple web form (see Figure 3) is provided where they fill the URL to be shortened and the needed metadata. Then, they are presented with a handle URL to use in their work, see Figure 2. The handle suffix is a combination of alphanumeric characters, dashes, underscores, full stops and other common characters.

For collaborating services or for advanced users REST API is also provided which allows the creation to be done programmatically. The idea is that the target service (the service featuring the user data queries) has a simple button or link on the query result page. The user simply clicks a button and is presented with a citable text that can be used in a paper.

To prevent abuse of the service and enforce accountable-usage, authorisation and authentication is needed. For the web interface, Shibboleth[6] (SAML) is used. Despite several drawbacks, this setup allows many academic users to sign in. For the rest, secret authentication tokens are generated. It should be clear from the data submitted who the author is; even if the request is coming from a service.

4. Conclusion

We have presented a URL shortener service - shortref.org - that can improve the impact of research data and the reproducibility of research results. It allows for short, persistent and easy-to-cite data queries which can be referenced from scientific papers.

It is based on the robust handle system resolution service that is being used for many millions of PIDs. The persistent identifier record itself contains all the metadata necessary to describe the query - date of creation, provenance, title and contact information and the final URL. In case the URL is not working, the metadata can be leveraged to identify the versions of software and data in time, or to contact either the creators of the PID or authors of the service. The architecture enables easy migration of the service completely transparent to the user.

[6]Shibboleth is among the world's most widely deployed federated identity solutions. https://shibboleth.net/

5. Acknowledgments

This work has been supported by the LINDAT/CLARIN project No. LM2015071 of the MEYS CR.

6. References

Kahn, R. and Wilensky, R. (2006). A framework for distributed digital object services. *International Journal on Digital Libraries*, 6(2):115–123.

Piwowar, H. A., Day, R. S., and Fridsma, D. B. (2007). Sharing detailed research data is associated with increased citation rate. *PloS one*.

Linking Language Resources and NLP papers

Gil Francopoulo, LIMSI, CNRS, Université Paris-Saclay + Tagmatica (France)
Joseph Mariani, LIMSI, CNRS, Université Paris-Saclay (France)
Patrick Paroubek, LIMSI, CNRS, Université Paris-Saclay (France)

Abstract

The Language Resources and Evaluation Map (LRE Map) is an accessible database on Language Resources based on records collected during the submission of several major Speech and Natural Language Processing (NLP) conferences, including the Language Resources and Evaluation Conferences (LREC). The NLP4NLP is a very large corpus of scientific papers in the field of Speech and Natural Language Processing covering a large number of conferences and journals in that field. In this article, we establish the link between those two elements in order to study the mention of the LRE Map resource names within the NLP4NLP corpus.

Keywords: Resource Citation, Named Entity Detection, Informetrics, Scientometrics, Text Mining, LRE Map.

1. Introduction

Our work is based on the hypothesis that names, in this case language resource names, correlate with the study, use and improvement of the given referred objects, in this case language resources. We believe that the automatic (and objective) detection is a step towards the improvement of the reliability of language resources as mentioned in [Branco 2013].

We already have an idea on how the resources are used in the recent venues of conferences such as Coling and LREC, as the LRE Map is built according to the resources declared by the authors of these conferences [Calzolari et al 2012]. But what about the other conferences and the other years? This is the subject of the present study.

2. Situation with respect to other studies

The approach is to apply NLP tools on texts about NLP itself, taking advantage of the fact that we have a good knowledge of the domain ourselves. Our work goes after the various studies presented and initiated in the Workshop entitled: "Rediscovering 50 Years of Discoveries in Natural Language Processing" on the occasion of ACL's 50th anniversary in 2012 [Radev et al 2013] where a group of researchers studied the content of the corpus recorded in the ACL Anthology [Bird et al 2008]. Various studies, based on the same corpus followed, for instance [Bordea et al 2014] on trend analysis and resulted in systems such as Saffron[1] or the Michigan Univ. web site[2]. Other studies were conducted by ourselves specifically on speech-related archives [Mariani et al 2013], and on the LREC archives [Mariani et al 2014a] but the target was to detect the terminology used within the articles, and the focus was not to detect resource names. More focused on the current workshop topic is the study conducted by the Linguistic Data Consortium (LDC) team whose goal was, and still is, to build a language resource (LR) database documenting the use of the LDC resources [Ahtaridis et al 2012]. At the time of the publication (i.e. 2012), the LDC team found 8,000 references and the problems encountered were documented in [Mariani et al 2014b].

3. Our approach

The general principle is to confront the names of the LRE Map with the newly collected NLP4NLP corpus. The process is as follows:

- Consider the archives of (most of) the NLP field,
- Take an entity name detector which is able to work with a given list of proper names,
- Use the LRE Map as the given list of proper names,
- Run the application and study the results.

4. Archives of a large part of the NLP field

The corpus is a large content of our own research field, i.e. NLP, covering both written and speech sub-domains and extended to a limited number of corpora, for which Information Retrieval and NLP activities intersect. This corpus was collected at IMMI-CNRS and LIMSI-CNRS (France) and is named NLP4NLP[3]. It currently contains 65,003 documents coming from various conferences and journals with either public or restricted access. This is a large part of the existing published articles in our field, apart from the workshop proceedings and the published books. Despite the fact that they often reflect innovative trends, we did not include workshops as they may be based on various reviewing processes and as the access to their content may sometimes be difficult. The time period spans from 1965 to 2015. Broadly speaking, and aside from the small corpora, one third comes from the ACL Anthology[4], one third from the ISCA Archive[5] and one third from IEEE[6].

[1] http://saffron.deri.ie
[2] http://clair.eecs.umich.edu/aan/index.php
[3] See www.nlp4nlp.org
[4] http://aclweb.org/anthology
[5] www.isca-speech.org/iscaweb/index.php/archive/online-archive
[6] https://www.ieee.org/index.html

The corpus follows the organization of the ACL Anthology with two parts in parallel. For each document, on one side, the metadata is recorded with the author names and the title. On the other side, the PDF document is recorded on disk in its original form. Each document is labeled with a unique identifier, for instance "lrec2000_1" is reified on the hard disk as two files: "lrec2000_1.bib" and "lrec2000_1.pdf". When recorded as an image, the PDF content is extracted by means of Tesseract OCR[7]. The automatic test leading to the call (or not) of the OCR is implemented by means of some PDFBox[8] API calls. For all the other documents, other PDFBox API calls are applied in order to extract the textual content. See [Francopoulo et al 2015] for more details about the extraction process as well as the solutions for some tricky problems like joint conferences management.

The majority (90%) of the documents come from conferences, the rest coming from journals. The overall number of words is 270M. Initially, the texts are in four languages: English, French, German and Russian. The number of texts in German and Russian is less than 0.5%. They are detected automatically and are ignored. The texts in French are a little bit numerous (3%), so they are kept with the same status as the English ones. This is not a problem because our tool is able to process English and French. The number of different authors is 48,894. The detail is presented in table 1.

5. Named Entity Detection

The aim is to detect a given list of names of resources, provided that the detection should be robust enough to recognize and link as the same entry some typographic variants such as "British National Corpus" vs "British National corpus" and more elaborated aliases like "BNC". Said in other terms, the aim is not to recognize some given raw character strings but also to link names together, a process often labeled as "entity linking" in the literature [Guo et al 2011][Moro et all 2014]. We use the industrial Java-based parser TagParser[9] [Francopoulo 2007] which, after a deep robust parsing for English and French, performs a named entity detection and then an entity linking processing. The system is hybrid, combining a statistical chunker, a large language specific lexicon, a multilingual knowledge base with a hand-written set of rules for the final selection of the named entities and their entity linking.

6. The LRE Map

The LRE Map is a freely accessible large database on resources dedicated to Natural Language Processing (NLP). The original feature of LRE Map is that the records are collected during the submission of different major NLP conferences[10]. These records were collected directly from the authors. We use the version of the LRE Map collected from 10 conferences from 2010 to 2012 within the EC FlaReNet project as described in [Mariani et al 2015].

The original version was a list of resource descriptions: this does not mean that this is a list of resource names which could be directly used in a recognition system, because what we need for each entry is a proper name, possibly associated with some alternate names. The number of entries was originally 4,396. Each entry has been defined with a headword like "British National Corpus" and some of them are associated with alternate names like "BNC". We further cleaned the data, by regrouping the duplicate entries, by omitting the version number which was associated with the resource name for some entries, and by ignoring the entries which were not labeled with a proper name but through a textual definition and those which had no name. Once cleaned, the number of entries is now 1,301, all of them with a different proper name. All the LRE Map entries are classified according to a very detailed set of resource types. We reduced the number of types to 5 broad categories: NLPCorpus, NLPGrammar, NLPLexicon, NLPSpecification and NLPTool, with the convention that when a resource is both a specification and a tool, the "specification" type is retained. An example is ROUGE which is both a set of metrics and a software package implementing those metrics, for which we chose the "specification" type.

7. Connection of LRE Map with TagParser

TagParser is natively associated with a large multilingual knowledge base made from Wikidata and Wikipedia and whose name is Global Atlas [Francopoulo et al 2013]. Of course, at the beginning, this knowledge base did not contain all the names of the LRE Map. Only 30 resource names were known like "Wikipedia" or "WordNet". During the preparation of the experiment, a data fusion has been applied between the two lists to incorporate the LRE Map into the knowledge base.

8. Running session and post-processing

The entity name detection is applied to the whole corpus on a middle range machine, i.e. one Xeon E3-1270V2 with 32Gb of memory. A post-processing is done in order to filter only the linked entities of the types: NLPCorpus, NLPGrammar, NLPLexicon, NLPSpecification and NLPTool. Then the results are gathered to compute a readable synthesis as an HTML file which is too big to be presented here, but the interested reader may consult the file "lremap.html" on www.nlp4nlp.org. Let's add that the whole computation takes 95 minutes.

[7] https://code.google.com/p/tesseract-ocr

[8] https://pdfbox.apache.org

[9] www.tagmatica.com

[10] As defined in https://en.wikipedia.org/wiki/LRE_Map

short name	# docs	format	long name	language	access to content	period	# venues
acl	4264	conference	Association for Computational Linguistics Conference	English	open access *	1979-2015	37
acmtslp	82	journal	ACM Transaction on Speech and Language Processing	English	private access	2004-2013	10
alta	262	conference	Australasian Language Technology Association	English	open access *	2003-2014	12
anlp	278	conference	Applied Natural Language Processing	English	open access *	1983-2000	6
cath	932	journal	Computers and the Humanities	English	private access	1966-2004	39
cl	776	journal	American Journal of Computational Linguistics	English	open access *	1980-2014	35
coling	3813	conference	Conference on Computational Linguistics	English	open access *	1965-2014	21
conll	842	conference	Computational Natural Language Learning	English	open access *	1997-2015	18
csal	762	journal	Computer Speech and Language	English	private access	1986-2015	29
eacl	900	conference	European Chapter of the ACL	English	open access *	1983-2014	14
emnlp	2020	conference	Empirical methods in natural language processing	English	open access *	1996-2015	20
hlt	2219	conference	Human Language Technology	English	open access *	1986-2015	19
icassps	9819	conference	IEEE International Conference on Acoustics, Speech and Signal Processing - Speech Track	English	private access	1990-2015	26
ijcnlp	1188	conference	International Joint Conference on NLP	English	open access *	2005-2015	6
inlg	227	conference	International Conference on Natural Language Generation	English	open access *	1996-2014	7
isca	18369	conference	International Speech Communication Association	English	open access	1987-2015	28
jep	507	conference	Journées d'Etudes sur la Parole	French	open access *	2002-2014	5
lre	308	journal	Language Resources and Evaluation	English	private access	2005-2015	11
lrec	4552	conference	Language Resources and Evaluation Conference	English	open access *	1998-2014	9
ltc	656	conference	Language and Technology Conference	English	private access	1995-2015	7
modulad	232	journal	Le Monde des Utilisateurs de L'Analyse des Données	French	open access	1988-2010	23
mts	796	conference	Machine Translation Summit	English	open access	1987-2015	15
muc	149	conference	Message Understanding Conference	English	open access *	1991-1998	5
naacl	1186	conference	North American Chapter of the ACL	English	open access *	2000-2015	11
paclic	1040	conference	Pacific Asia Conference on Language, Information and Computation	English	open access *	1995-2014	19
ranlp	363	conference	Recent Advances in Natural Language Processing	English	open access *	2009-2013	3
sem	950	conference	Lexical and Computational Semantics / Semantic Evaluation	English	open access *	2001-2015	8
speechc	593	journal	Speech Communication	English	private access	1982-2015	34
tacl	92	journal	Transactions of the Association for Computational Linguistics	English	open access *	2013-2015	3
tal	177	journal	Revue Traitement Automatique du Langage	French	open access	2006-2015	10
taln	1019	conference	Traitement Automatique du Langage Naturel	French	open access *	1997-2015	19
taslp	6612	journal	IEEE/ACM Transactions on Audio, Speech and Language Processing	English	private access	1975-2015	41
tipster	105	conference	Tipster DARPA text program	English	open access *	1993-1998	3
trec	1847	conference	Text Retrieval Conference	English	open access	1992-2015	24
cell total	67937[11]					1965-2015	577

Table 1: Detail of NLP4NLP, with the convention that an asterisk indicates that the corpus is in the ACL Anthology.

9. Global counting over the whole history

In order to avoid any misleading, we adopt the same conventions as in our other studies, as follows:

- the number of occurrences of a resource name is N when the name is mentioned N times in a document,
- the number of presences of a resource name is 1 when the name is mentioned M times in a document, with M > 0.

We think that the number of presences is a better indicator than the number of occurrences because a resource name may be mentioned several times in a paper for wording reasons, for instance in the body and the conclusion, but

[11] In the general counting, for a joint conference (which is a rather infrequent situation), the paper is counted once (giving 65,003), so the sum of all cells in the table is slightly more important (giving 67,937). Similarly, the number of venues is 558 when the joint conferences are counted once, but 577 when all venues are counted.

what is important is whether the resource is used or not. Year after year, the number of documents per year increases, as presented in figure 1 with the orange line. The number of presences of Language Resources also increases as presented with the blue line.

That means that year after year, more and more LR are mentioned, both as raw counting and as number of presences per document. But we must not forget that there is a bias which boosts the effect: the point is that only recent and permanent resources are recorded in the LRE Map. For instance a resource invented in the 80s' and not used since the creation of the LRE Map in 2010 is not recorded in the LRE Map and will therefore be ignored in our analysis. We see that the number of the presences of Language Resource gets equal to the number of documents in 2006-2007 (it means that on average a Language Resource is mentioned in each paper, as it also appears in figure 2). This period may therefore be considered as the time when the research paradigm in Language Processing turned from mostly model-driven to mostly data-driven. The number of presences then gets even larger than the number of documents.

10. Global top 10 over the history

Over the whole history, when only the top 10 resources are considered, the result is as follows in table 2, ordered by the number of presences in decreasing order. The evolution over the history is presented in figure 3.

There was no mention until 1989, as the earliest LR, TIMIT, appeared at that time. We however see that TIMIT is still much in use after 26 years. The evolution from 1989 until 2015 for these top 10 resources shows for instance that during the period 2004-2011 the resource name "WordNet" was more popular than "Wikipedia", but since 2011, it is the contrary. We can notice also the ridges on even years due to some conferences related to Language Resources that are biennial, such as LREC and Coling on even years.

11. Top 10 for each year

Another way to present the results is to compute a top 10 for each year, as in table 3.

Resource	Type	# pres.	# occur.	First authors mentioning the LR	First corpora mentioning the LR	First year of mention	Last year	Rank
WordNet	NLPLexicon	4203	29079	Daniel A Teibel, George A Miller	hlt	1991	2015	1
Timit	NLPCorpus	3005	11853	Andrej Ljolje, Benjamin Chigier, David Goodine, David S Pallett, Erik Urdang, Francine R Chen, George R Doddington, H-W Hon, Hong C Leung, Hsiao-Wuen Hon, James R Glass, Jan Robin Rohlicek, Jeff Shrager, Jeffrey N Marcus, John Dowding, John F Pitrelli, John S Garofolo, Joseph H Polifroni, Judith R Spitz, Julia B Hirschberg, Kai-Fu Lee, L G Miller, Mari Ostendorf, Mark Liberman, Mei-Yuh Hwang, Michael D Riley, Michael S Phillips, Robert Weide, Stephanie Seneff, Stephen E Levinson, Vassilios V Digalakis, Victor W Zue	hlt, isca, taslp	1989	2015	2
Wikipedia	NLPCorpus	2824	20110	Ana Licuanan, J H Xu, Ralph M Weischedel	trec	2003	2015	3
Penn Treebank	NLPCorpus	1993	6982	Beatrice Santorini, David M Magerman, Eric Brill, Mitchell P Marcus	hlt	1990	2015	4
Praat	NLPTool	1245	2544	Carlos Gussenhoven, Toni C M Rietveld	isca	1997	2015	5
SRI Language Modeling Toolkit	NLPTool	1029	1520	Dilek Z Hakkani-Tür, Gökhan Tür, Kemal Oflazer	coling	2000	2015	6
Weka	NLPTool	957	1609	Douglas A Jones, Gregory M Rusk	coling	2000	2015	7
Europarl	NLPCorpus	855	3119	Daniel Marcu, Franz Josef Och, Grzegorz Kondrak, Kevin Knight, Philipp Koehn	acl, eacl, hlt, naacl	2003	2015	8
FrameNet	NLPLexicon	824	5554	Beryl T Sue Atkins, Charles J Fillmore, Collin F Baker, John B Lowe, Susanne Gahl	acl, coling, lrec	1998	2015	9
GIZA++	NLPTool	758	1582	David Yarowsky, Grace Ngai, Richard Wicentowski	hlt	2001	2015	10

Table 2: Top 10 most mentioned resources over the history

Year	# pres.of LR	# doc. in the year	Top10 cited resources (ranked)
1965	7	24	C-3, LLL, LTH, OAL, Turin University Treebank
1966	0	7	
1967	6	54	General Inquirer, LTH, Roget's Thesaurus, TFB, TPE
1968	3	17	General Inquirer, Medical Subject Headings
1969	4	24	General Inquirer, Grammatical Framework GF
1970	2	18	FAU, General Inquirer
1971	0	20	
1972	2	19	Brown Corpus, General Inquirer
1973	7	80	ANC Manually Annotated Sub-corpus, Grammatical Framework GF, ILF, Index Thomisticus, Kontrast, LTH, PUNKT
1974	8	25	General Inquirer, Brown Corpus, COW, GG, LTH
1975	15	131	C-3, LTH, Domain Adaptive Relation Extraction, ILF, Acl Anthology Network, BREF, LLL, Syntax in Elements of Text, Unsupervised incremental parser
1976	13	136	Grammatical Framework GF, LTH, C-3, DAD, Digital Replay System, Domain Adaptive Relation Extraction, General Inquirer, Perugia Corpus, Syntax in Elements of Text, Talbanken
1977	8	141	Grammatical Framework GF, Corpus de Referencia del Español Actual, Domain Adaptive Relation Extraction, GG, LTH, Stockholm-Umeå corpus
1978	16	155	Grammatical Framework GF, C-3, General Inquirer, Digital Replay System, ILF, LLL, Stockholm-Umeå corpus, TDT
1979	23	179	Grammatical Framework GF, LLL, LTH, C-3, C99, COW, CTL, ILF, ItalWordNet, NED
1980	38	307	Grammatical Framework GF, C-3, LLL, LTH, ANC Manually Annotated Sub-corpus, Acl Anthology Network, Automatic Statistical SEmantic Role Tagger, Brown Corpus, COW, CSJ
1981	33	274	C-3, Grammatical Framework GF, LTH, Index Thomisticus, CTL, JWI, Automatic Statistical SEmantic Role Tagger, Brown Corpus, Glossa, ILF
1982	40	364	C-3, LLL, LTH, Brown Corpus, GG, ILF, Index Thomisticus, Arabic Gigaword, Arabic Penn Treebank, Automatic Statistical SEmantic Role Tagger
1983	59	352	Grammatical Framework GF, C-3, LTH, GG, LLL, Unsupervised incremental parser, LOB Corpus, OAL, A2ST, Arabic Penn Treebank
1984	55	353	LTH, Grammatical Framework GF, PET, LLL, C-3, CLEF, TLF, Arabic Penn Treebank, Automatic Statistical SEmantic Role Tagger, COW
1985	53	384	Grammatical Framework GF, LTH, C-3, LOB Corpus, Brown Corpus, Corpus de Referencia del Español Actual, LLL, DCR, MMAX, American National Corpus
1986	92	518	LTH, C-3, LLL, Digital Replay System, Grammatical Framework GF, DCR, JRC Acquis, Nordisk Språkteknologi, Unsupervised incremental parser, OAL
1987	63	669	LTH, C-3, Grammatical Framework GF, DCR, Digital Replay System, LOB Corpus, CQP, EDR, American National Corpus, Arabic Penn Treebank
1988	105	546	C-3, LTH, Grammatical Framework GF, Digital Replay System, DCR, Brown Corpus, FSR, ISOcat Data Category Registry, LOB Corpus, CTL
1989	145	965	Grammatical Framework GF, Timit, LTH, LLL, C-3, Brown Corpus, Digital Replay System, LTP, DCR, EDR
1990	175	1277	Timit, Grammatical Framework GF, LTH, C-3, LLL, Brown Corpus, GG, LTP, ItalWordNet, JRC Acquis
1991	240	1378	Timit, LLL, C-3, LTH, Grammatical Framework GF, Brown Corpus, Digital Replay System, LTP, GG, Penn Treebank
1992	361	1611	Timit, LLL, LTH, Grammatical Framework GF, Brown Corpus, C-3, Penn Treebank, WordNet, GG, ILF
1993	243	1239	Timit, WordNet, Penn Treebank, Brown Corpus, EDR, LTP, User-Extensible Morphological Analyzer for Japanese, BREF, Digital Replay System, James Pustejovsky
1994	292	1454	Timit, LLL, WordNet, Brown Corpus, Penn Treebank, C-3, Digital Replay System, JRC Acquis, LTH, Wall Street Journal Corpus
1995	290	1209	Timit, LTP, WordNet, Brown Corpus, Digital Replay System, LLL, Penn Treebank, Grammatical Framework GF, TEI, Ntimit
1996	394	1536	Timit, LLL, WordNet, Brown Corpus, Digital Replay System, Penn Treebank, Centre for Spoken Language Understanding Names, LTH, EDR, Ntimit
1997	428	1530	Timit, WordNet, Penn Treebank, Brown Corpus, LTP, HCRC, Ntimit, BREF, LTH, British National Corpus
1998	883	1953	Timit, WordNet, Penn Treebank, Brown Corpus, EuroWordNet, British National Corpus, Multext, EDR, LLL, PAROLE
1999	481	1603	Timit, WordNet, Penn Treebank, TDT, Maximum Likelihood Linear Regression, EDR, Brown Corpus, TEI, LTH, LLL
2000	842	2271	Timit, WordNet, Penn Treebank, British National Corpus, PAROLE, Multext, EuroWordNet, Maximum Likelihood Linear Regression, TDT, Brown Corpus
2001	648	1644	WordNet, Timit, Penn Treebank, Maximum Likelihood Linear Regression, TDT, Brown Corpus, CMU Sphinx, Praat, LTH, British National Corpus
2002	1105	2174	WordNet, Timit, Penn Treebank, Praat, EuroWordNet, British National Corpus, PAROLE, NEGRA, TDT, Grammatical Framework GF
2003	1067	1984	Timit, WordNet, Penn Treebank, AQUAINT, British National Corpus, AURORA, FrameNet, Praat, SRI Language Modeling Toolkit, OAL
2004	2066	2712	WordNet, Timit, Penn Treebank, FrameNet, AQUAINT, British National Corpus, EuroWordNet, Praat, PropBank, SemCor
2005	2006	2355	WordNet, Timit, Penn Treebank, Praat, AQUAINT, PropBank, British National Corpus, SRI Language Modeling Toolkit, MeSH, TDT
2006	3532	2794	WordNet, Timit, Penn Treebank, Praat, PropBank, AQUAINT, FrameNet, GALE, EuroWordNet, British National Corpus
2007	2937	2489	WordNet, Timit, Penn Treebank, Praat, SRI Language Modeling Toolkit, Wikipedia, GALE, GIZA++, SemEval, AQUAINT
2008	4007	3078	WordNet, Wikipedia, Timit, Penn Treebank, GALE, PropBank, Praat, FrameNet, SRI Language Modeling Toolkit, Weka
2009	3729	2637	WordNet, Wikipedia, Timit, Penn Treebank, Praat, SRI Language Modeling Toolkit, GALE, Europarl, Weka, GIZA++
2010	5930	3470	WordNet, Wikipedia, Penn Treebank, Timit, Europarl, Praat, FrameNet, SRI Language Modeling Toolkit, GALE, GIZA++
2011	3859	2957	Wikipedia, WordNet, Timit, Penn Treebank, Praat, SRI Language Modeling Toolkit, Weka, GIZA++, Europarl, GALE
2012	6564	3419	Wikipedia, WordNet, Timit, Penn Treebank, Europarl, Weka, Praat, SRI Language Modeling Toolkit, GIZA++, FrameNet
2013	5669	3336	Wikipedia, WordNet, Timit, Penn Treebank, Weka, SRI Language Modeling Toolkit, Praat, GIZA++, Europarl, SemEval
2014	6700	3817	Wikipedia, WordNet, Timit, Penn Treebank, Praat, Weka, SRI Language Modeling Toolkit, SemEval, Europarl, FrameNet
2015	5597	3314	Wikipedia, WordNet, Timit, SemEval, Penn Treebank, Praat, Europarl, Weka, SRI Language Modeling Toolkit, FrameNet

Table 3: Top 10 mentioned resources per year

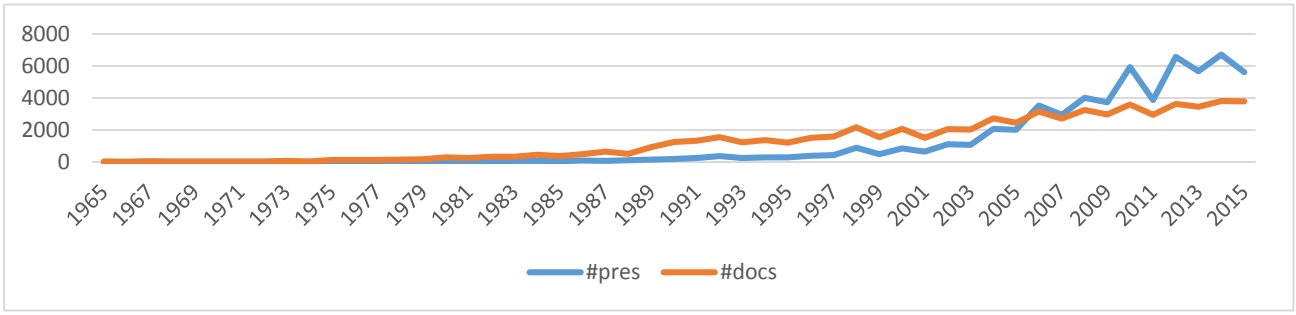


Figure 1: Presence of LR and total number of documents

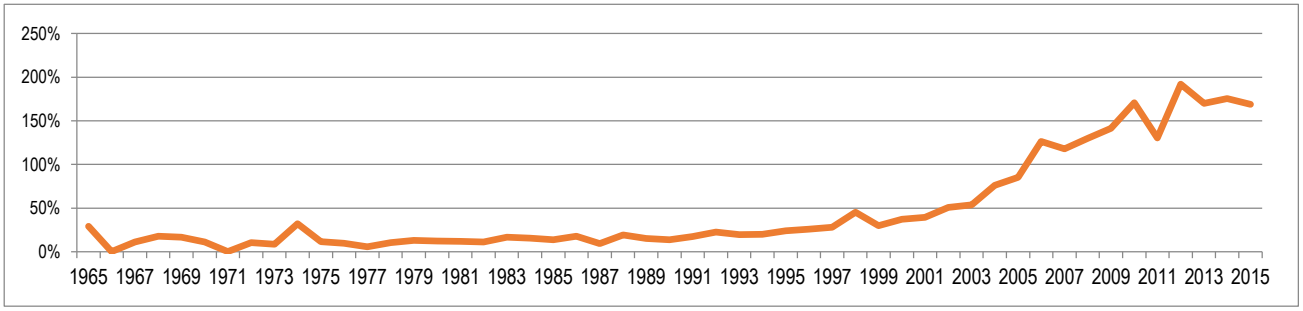


Figure 2: Percentage of LR presence in papers

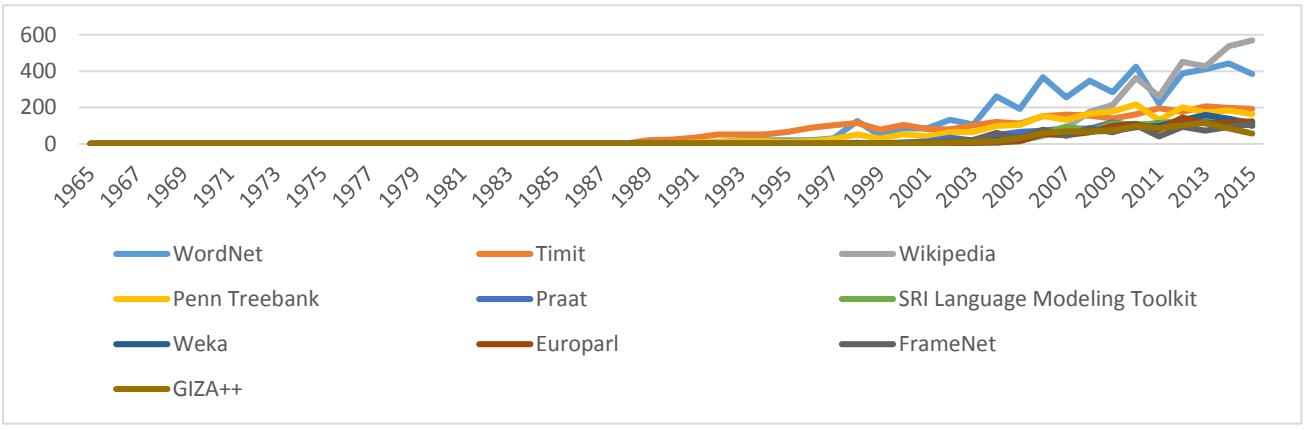


Figure 3: Evolution of the 10 Top LR presences over time

A different way to present the evolution of the terms is to compute a tag cloud at different points in time, for instance every 10 years in 1994, 2004 and 2014 by means of the site Tag Crowd [12]. Let's note that we chose the option to consider 2014 instead of 2015, as LREC and COLING did not occur in 2015.

Figure 4: Tagcloud for 1994

Figure 5: Tag cloud for 2004

We see in those figures the sustainable interest over the years for resources such as TIMIT, Wordnet or Penn Treebank. The relative popularity of others such as the Brown Corpus or the British National Corpus decreased over time, while it increased for others such as Wikipedia or Praat, which came to the forefront

[12] http://tagcrowd.com/

Figure 6: Tag cloud for 2014

12. Targeted study on “wordnet”

Instead of considering the whole set of names, another way to proceed is to select a name, starting from its first mention and to present its evolution, year after year. Let’s consider “WordNet”, starting in 1991 in the figure 7.

Another interesting view is the display the propagation of a specific term from a conference to another by means of a propagation matrix to be read from the top to the bottom. For instance, the first mention of “WordNet” (in our field) was issued in the Human Language Technology (HLT) conference in 1991 (first line). The term propagated in the NLP community through MUC, ACL, TREC and COLING in 1992, then in TIPSTER in 1993 and in the Speech community in 1994 (through the ISCA conference and the Computer Speech and Language journal), as presented in the following matrix of table 4, with the convention that the striped lines indicate that the corresponding corpus doesn’t exist in NLP4NLP, in case of biennal conferences, for example.

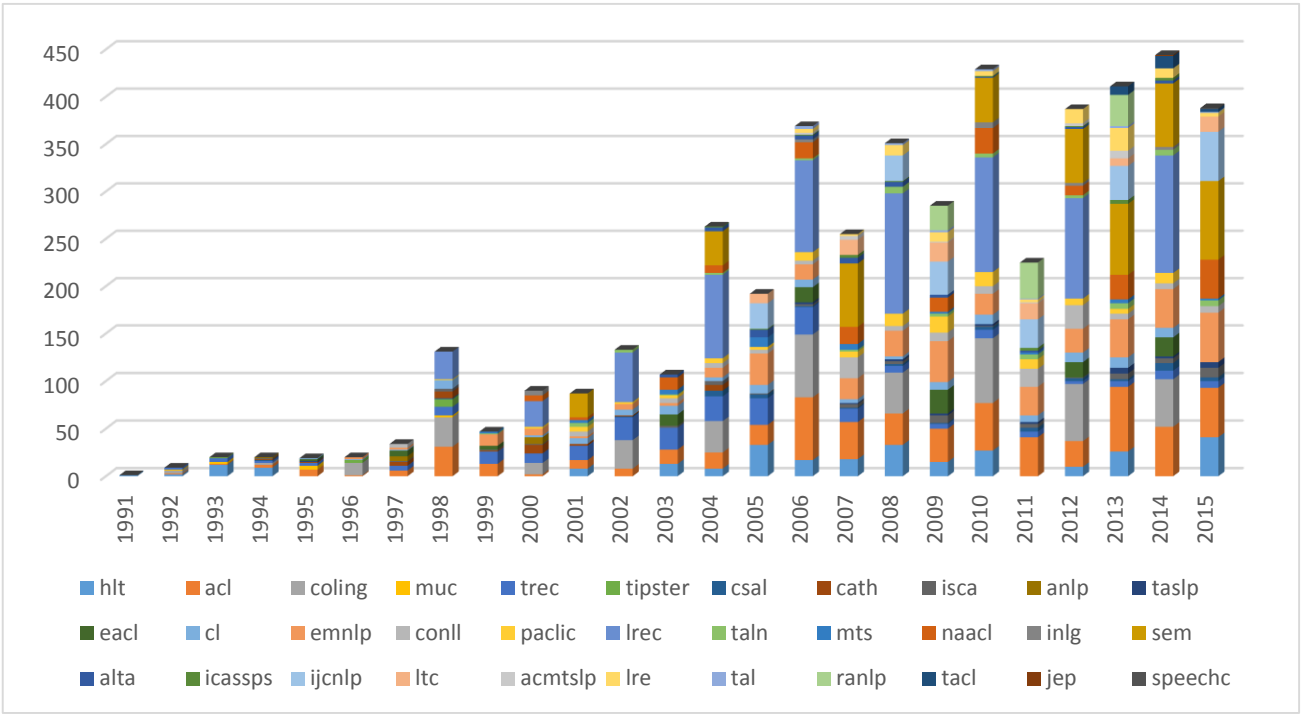


Figure 7: Evolution of "WordNet" presence over time

	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
hlt																									
muc																									
acl																									
trec																									
coling																									
tipster																									
anlp																									
isca																									
csal																									
cath																									
cl																									
eacl																									
taslp																									
emnlp																									
conll																									
paclic																									
lrec																									
taln																									
mts																									
inlg																									
naacl																									
sem																									
icassps																									
alta																									
ijcnlp																									
ltc																									
tal																									
lre																									
acmtslp																									
ranlp																									
tacl																									
jep																									
speechc																									

Table 4: Propagation matrix for “WordNet”

13. Targeted study on “Wikipedia”

Let’s see the evolution of another term like “Wikipedia”, starting in 2003, as follows:

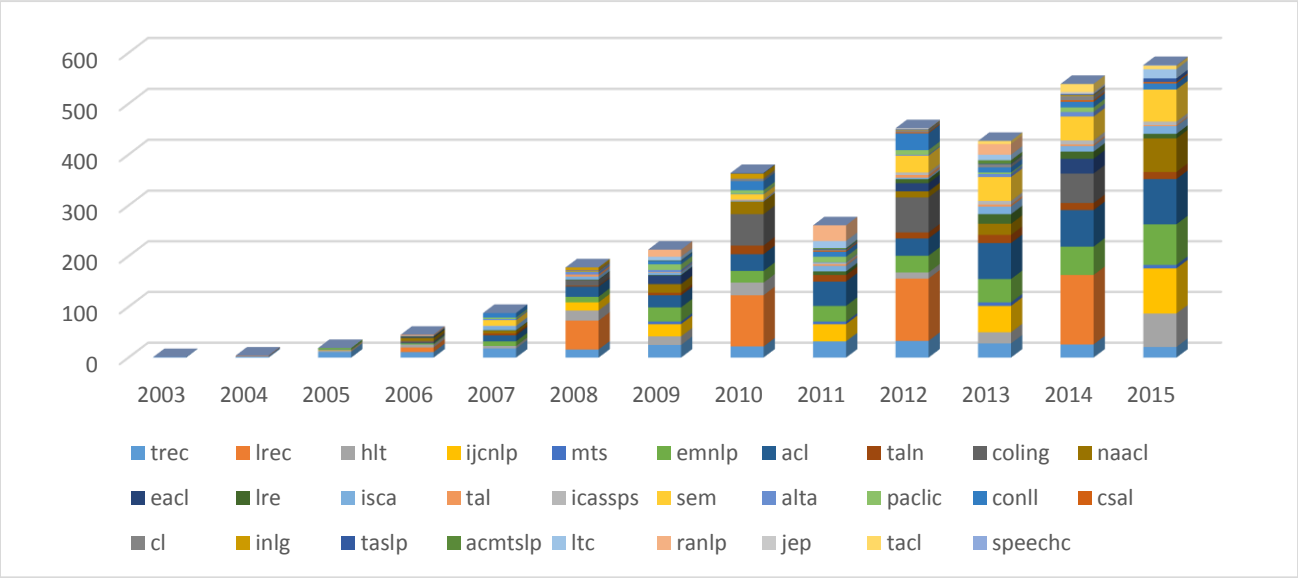


Figure 8: Evolution of "Wikipedia" presence over time

14. Conclusion and Perspective

To our knowledge, this study is the first which matches the content of the LRE Map with the scientific papers published in our field. Beforehand the LRE Map resources were related to the papers of conferences such as Coling and LREC, as the authors were invited to declare these resources during the different paper submission phases, but we had no idea on how these resources were used in other conferences and in other years. Of course, our approach does not cover all the names over the history. For instance a resource invented in the 80s' and not used anymore since 2010 is not recorded in the LRE Map and will therefore be ignored in our analysis. However, we see that Language Resources are more and more used nowadays, and that on average more than one Language Resources is cited in a conference or journal paper. We now plan to consider measuring a resource innovation impact factor for our various sources, conferences and journals: which are the sources where new resources are first mentioned that will later spread in other publications?

14. Acknowledgements

We'd like to thank Wolfgang Hess for the ISCA archive, Douglas O'Shaughnessy, Denise Hurley, Rebecca Wollman and Casey Schwartz for the IEEE data, Nicoletta Calzolari, Helen van der Stelt and Jolanda Voogd for the LRE Journal articles, Olivier Hamon and Khalid Choukri for the LREC proceedings, Nicoletta Calzolari, Irene Russo, Riccardo Del Gratta, Khalid Choukri for the LRE Map, Min-Yen Kan for the ACL Anthology, Florian Boudin for the TALN proceedings and Ellen Voorhees for the TREC proceedings.

15. Bibliographic References

Ahtaridis Eleftheria, Cieri Christopher, DiPersio Denise (2012), LDC Language Resource Database: Building a Bibliographic Database, Proceedings of LREC 2012, Istanbul, Turkey.

Bird Steven, Dale Robert, Dorr Bonnie J, Gibson Bryan, Joseph Mark T, Kan Min-Yen, Lee Dongwon, Powley Brett, Radev Dragomir R, Tan Yee Fan (2008), The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics, Proceedings of LREC, Marrakech, Morocco.

Bordea Georgeta, Buitelaar Paul, Coughlan Barry (2014), Hot Topics and schisms in NLP: Community and Trend Analysis with Saffron on ACL and LREC Proceedings, Proceedings of LREC 2014, 26-31 May 2014, Reykjavik, Iceland.

Branco Antonio (2013), Reliability and Meta-reliability of language resources : ready to initiate the integrity debate ? TLT12 COS, Centre for Open Science.

Calzolari Nicoletta, Del Gratta Riccardo, Francopoulo Gil, Mariani Joseph, Rubino Francesco, Russo Irene, Soria Claudia (2012), The LRE Map. Harmonising Community Descriptions of Resources, Proceedings of LREC, Istanbul, Turkey.

Francopoulo Gil (2007), TagParser : well on the way to ISO-TC37 conformance. ICGL (International Conference on Global Interoperability for Language Resources), Hong Kong, PRC.

Francopoulo Gil, Marcoul Frédéric, Causse David, Piparo Grégory (2013), Global Atlas: Proper Nouns, from Wikipedia to LMF, in LMF Lexical Markup Framework (Francopoulo, ed), ISTE Wiley.

Francopoulo Gil, Mariani Joseph, Paroubek Patrick (2015), NLP4NLP: the cobbler's children won't go unshod, in D-Lib Magazine : The magazine of Digital Library Research[13].

Guo Yuhang, Che Wanxiang, Liu Ting, Li Sheng (2011), A Graph-based Method for Entity Linking, International Joint Conference on NLP, Chiang Mai, Thailand.

Mariani Joseph, Paroubek Patrick, Francopoulo Gil, Delaborde Marine (2013), Rediscovering 25 Years of Discoveries in Spoken Language Processing: a Preliminary ISCA Archive Analysis, Proceedings of Interspeech 2013, 26-29 August 2013, Lyon, France.

Mariani Joseph, Paroubek Patrick, Francopoulo Gil, Hamon Olivier (2014a), Rediscovering 15 Years of Discoveries in Language Resources and Evaluation: The LREC Anthology Analysis, Proceedings of LREC 2014, 26-31 May 2014, Reykjavik, Iceland.

Mariani Joseph, Cieri Christopher, Francopoulo Gil, Paroubek Patrick, Delaborde Marine (2014b), Facing the Identification Problem in Language-Related Scientific Data Analysis, Proceedings of LREC 2014, 26-31 May 2014, Reykjavik, Iceland.

Mariani Joseph, Francopoulo Gil (2015), Language Matrices and a Language Resource Impact Factor, in Language Production, Cognition, and the lexicon (Nuria Gala, Reihard Rapp, Gemma Bel-Enguix editors), Springer.

Moro Andrea, Raganato Alessandro, Navigli Roberto (2014), Entity Linking meets Word Sense Disambiguation : a Unified Approach, Transactions of the Association for Computational Linguistics.

Radev Dragomir R, Muthukrishnan Pradeep, Qazvinian Vahed, Abu-Jbara, Amjad (2013), The ACL Anthology Network Corpus, Language Resources and Evaluation 47: 919–944.

[13]dlib.org/dlib/november15/francopoulo/11francopoulo.html