Proceedings of the Fifth
Web as Corpus Workshop
(WAC5)

Edited by Iñaki Alegria, Igor Leturia, Serge Sharoff

Pre-SEPLN Workshop
Donostia-San Sebastián, Basque Country, Spain
7 September, 2009

Published in San Sebastian by Elhuyar Fundazioa - 2009
ISBN: 978-84-92457-05-2

Organised under the auspices of ACL SIGWAC, a Special Interest Group on Web As Corpus of the Association for Computational Linguistics.

Workshop Organisers

Iñaki Alegria, University of the Basque Country
Adam Kilgarriff, Lexical Computing Ltd.
Igor Leturia, Elhuyar Fundazioa
Serge Sharoff, University of Leeds

Programme Committee

Silvia Bernardini, U of Bologna, Italy
Jesse de Does, INL, Netherlands
Katrien Depuydt, INL, Netherlands
Stefan Evert, U of Osnabrück, Germany
Cédrick Fairon, UCLouvain, Belgium
William Fletcher, U.S. Naval Academy, USA
Gregory Grefenstette, Commissariat à l'Énergie Atomique, France
Katja Hofmann, U of Amsterdam, Netherlands
Adam Kilgarriff, Lexical Computing Ltd, UK
Igor Leturia, Elhuyar Fundazioa, Basque Country, Spain
Preslav Nakov, National U of Singapore
Phil Resnik, U of Maryland, College Park, USA
Kevin Scannell, Saint Louis U, USA
Gilles-Maurice de Schryver, U Gent, Belgium
Klaus Schulz, LMU München, Germany
Serge Sharoff, U of Leeds, UK
Eros Zanchetta, U of Bologna, Italy

Invited speaker

Dekang Lin, Google Inc

Local organisation

Antton Gurrutxaga, Elhuyar Fundazioa
Rakel Lopez, Elhuyar Fundazioa
Maddalen Lopez de Lacalle, Elhuyar Fundazioa
Iker Manterola, Elhuyar Fundazioa
Eli Pociello, Elhuyar Fundazioa
Iñaki San Vicente, Elhuyar Fundazioa
Xabier Saralegi, Elhuyar Fundazioa

Workshop Programme

9.15 – 9.30	Welcome & Introduction
Session 1	**Collecting Web corpora** (1)
9.30 – 10.00	*Jonathan Howell and Mats Rooth.* Web Harvest of Minimal Intonational Pairs
10.00 – 10.30	*Marco Brunello.* The creation of free linguistic corpora from the web
10.30 – 11.00	Coffee break
Session 2	**Aspects of Web processing** (1)
11.00 – 11.30	*Eugenie Giesbrecht and Stefan Evert.* Part-of-Speech (POS) Tagging - a Solved Task? An Evaluation of POS Taggers for the Web as Corpus
11.30 – 12.00	*Matthias Wendt, Christoph Büscher, Christian Herta, Steffen Kemmerer, Walter Tietze, Manuel Messner, Martin Gerlach and Holger Düwiger.* Extracting domain terminologies from the World Wide Web
12.00 – 13.00	*Dekang Lin.* Unsupervised acquisition of lexical knowledge from the Web
13.00 – 15.00	Lunch break
Session 3	**Collecting Web corpora** (2)
15.00 – 15.30	*Johannes Steger and Egon Stemle.* The Architecture for Unified Processing of Web Content
15.30 – 16.00	*Igor Leturia, Iñaki San Vicente and Xabier Saralegi.* Search engine based approaches for collecting domain-specific Basque-English comparable corpora from the Internet
16.00 – 16.30	Coffee break
Session 4	**Aspects of Web processing** (2)
16.30 – 17.00	*Joel Tetreault and Martin Chodorow.* Examining the Use of Region Web Counts for ESL Error Detection Missing reviews: Péter Halácsy, Adam Kilgarriff
17.00 – 17.30	*Kristin Davidse and Emeline Doyen.* Using Internet data for the study of language change: a comparative study of the grammaticalized uses of French genre in teenage and adult forum data
17.30 – 18.00	*Nabil Hathout, Franck Sajous and Ludovic Tanguy.* Looking for French deverbal nouns in an evolving Web (a short history of WAC)
18.00 - 18.30	General discussion, wrap-up & conclusion

Contents

Preface **7**

1 The creation of free linguistic corpora from the web
Marco Brunello **9**

2 Using Internet data for the study of language change: a comparative study of the grammaticalized uses of French genre in teenage and adult forum data
Kristin Davidse and Emeline Doyen **17**

3 Part-of-Speech (POS) Tagging - a Solved Task? An Evaluation of POS Taggers for the Web as Corpus
Eugenie Giesbrecht and Stefan Evert **27**

4 Looking for French deverbal nouns in an evolving Web (a short history of WAC)
Nabil Hathout, Franck Sajous and Ludovic Tanguy **37**

5 Web Harvest of Minimal Intonational Pairs
Jonathan Howell and Mats Rooth **45**

6 Search engine based approaches for collecting domain-specific Basque-English comparable corpora from the Internet
Igor Leturia, Iñaki San Vicente and Xabier Saralegi **53**

7 The Architecture for Unified Processing of Web Content
Johannes Steger and Egon Stemle **63**

8 Examining the Use of Region Web Counts for ESL Error Detection
Joel Tetreault and Martin Chodorow **71**

9 Extracting domain terminologies from the World Wide Web
Matthias Wendt, Christoph Büscher, Christian Herta, Steffen Kemmerer, Walter Tietze, Manuel Messner, Martin Gerlach and Holger Düwiger **79**

Preface

> By this art you may contemplate the variations of the 23 letters.
> Robert Burton (1621) *The Anatomy of Melancholy*

> The universe (which others call the Library) is composed of an indefinite and perhaps infinite number of hexagonal galleries, with vast air shafts between, surrounded by very low railings. Like all men of the Library, I have traveled in my youth; I have wandered in search of a book, perhaps the catalogue of catalogues; now that my eyes can hardly decipher what I write, I am preparing to die just a few leagues from the hexagon in which I was born. Once I am dead, there will be no lack of pious hands to throw me over the railing; my grave will be the fathomless air; my body will sink endlessly and decay and dissolve in the wind generated by the fall, which is infinite. I say that the Library is unending.
> Jorge Luis Borges (1941) *The Library of Babel*, trans. by James E. Irby

The pessimistic viewpoint presented by Borges echoes frustration often felt from attempts to get reasonable content from the web. The results returned for a query are often different from the results for the same query issued five minutes ago. Webpages get removed or renamed. They violate standards, use a mix of encodings, fool our part-of-speech taggers or parsers. The sheer variety of the Web makes any classification task impossible or next to impossible. Manfred Görlach in his *Text types and the history of English* lists more than 2,000 traditional genres without making a claim that this list covers printed genres exhaustively. The Web adds new text types to this inventory, such as homepages and blogs, wiki pages and participatory news articles, online shops and FAQs. At the moment we do not know precisely what is there, but we are gradually getting better understanding of how to get data from the Web and how to study our catch.

Even if, unlike the Library of Babel, it does not contain a text in *a Samoyedic Lithuanian dialect of Guarani, with classical Arabian inflections*, the range of languages is quite considerable. At the workshop we have presenters studying Basque, French, German, Italian, as well as Chinese, French and Russian dialects of English, i.e. studying errors made by respective language learners.

We are grateful to the University of the Basque Country and to the Government of the Basque Country for funding, SEPLN for hosting, Elhuyar Foundation for helping with the organization and the city of Donostia-San Sebastián for the nice venue.

Iñaki Alegria, Igor Leturia, Serge Sharoff

The creation of free linguistic corpora from the web

Marco Brunello
Università degli Studi di
Padova
Palazzo Maldura, via Beato
Pellegrino, 1, Padova
brunez@email.it

Abstract

This paper shows how it's possible to build free corpora from the web using documents released under Creative Commons licenses.

1 Introduction

Corpus linguistics originated at the end of the sixties, and it grew in importance in the last decades of the twentieth century, mainly because of one reason: the development of computer technologies. In fact, it was inevitable that corpus linguistics would encounter the biggest source of texts in electronic format that had ever existed: the world wide web. Internet is very precious for computational linguists, as it has some undeniable advantages: it is the largest collection of texts, always with recent and up-to-date data, and all the texts on the web are still in machine-readable form. But the discussion about the web as a corpus is still unresolved, and there are many opinions about this; let us now consider some of these, showing the various – and sometimes even contrasting – points of view.

The first one is Vegnaduzzo (2007). This study shows how to use automated searches on search engines for linguistic purposes, and obtain two kinds of data: frequency data (by watching the number of results of our searches we can determine measures of lexical association) and textual data (by watching the section “similar searches” we can find links of a linguistic nature between words). This method is very simple, and despite its validity it cannot be used for more complex linguistic purposes. This is due to the commercial – and not linguistic – nature of the search engines; the solution can be through the personalization of the existing search engines, by creating new tools implemented with additional search possibilities according to the linguistic purpose: some examples are *WebCorp*[1] and *KwiCfinder*[2]. This method considers the web “as a corpus surrogate” (Bernardini *et al.* 2006), but for some reasons (like the non-reproducibility of the results given by search engines (Lüdeling *et al.* 2007), this cannot be considered the best solution for using the web as a resource in a linguistic way. Another solution is to use the web only as a source of data for the creation of standalone corpora. This can be done by search methods deliberately developed for every purpose we want to pursue; a good example can be the WaCky project[3], developed by a community of computational linguists who created three great corpora of Italian, English and German (respectively *itWaC*, *ukWaC* and *deWaC*), entirely built with documents taken from web pages. Most probably this is one of the best way to make web corpora, because the use of automated works on search engines is minimal, but we should not forget a stronger application of automated searches and automated url selections; this method is for sure less careful and may require a lot of data-cleaning, but it's the best choice when we need a single-use corpus, for example, and there is limited time or economic resources. A good instrument for this job is BootCaT[4] (Baroni e Bernardini 2004): a toolkit composed by various tools, one for every step of the building of corpora, based on automated searches on Google or Yahoo!. There we will give a more detailed de-

[1] www.webcorp.org.uk

[2] www.kwicfinder.com

[3] Acronym of ***W**eb-**a**s-**C**orpus **k**ool **y**nitiative*. Online at http://wacky.sslmit.unibo.it

[4] Acronym of ***Boot**strapping **C**orpora **a**nd **T**erms*. Online at http://sslmit.unibo.it/~baroni/bootcat.html.

scription of this toolkit later, when it is shown how it was used; now we must consider another fundamental problem about the use of documents recovered from the web: the copyright.

2 Copyright and web-corpora: problems and solutions

The problem of the copyright on the text of which a corpora is composed isn't obviously related only to the web corpora, but it's more concerned with these because of the extreme ease in reproducing data in electronic format (and the web only has data in electronic format), especially nowadays when everybody, thanks to a simple personal computer, has all the necessary instruments to make copies of every kind of multimedial material: texts, audio tracks, images, videos etc., and a law stating “all rights reserved” cannot prevent the reproduction of a file found on Internet.

With reference to corpus linguistics, the problem is real and felt: as we said, the web is a very large source of textual data for the creation of corpora, and notwithstanding all the good intentions of a researcher that collects web data for building a corpus in the name of *fair use* (not committing acts of piracy), redistribute data taken from the web without the permission of their creator is illegal. The problem isn't concerned only with web corpora: we have various examples when the use and distribution of corpora is strictly limited to non-commercial uses or subject to the payment of royalties (Allora e Barbera 2007). In these examples, we can see that the normal copyright rules apply, but talking about non-commercial uses of the corpora, these distributors' guidelines are more restrictive[5] or aren't very light, and sometimes the question is tackled leaving large margins of ambiguity[6]. And, as we said about web corpora, the problem may be more serious, because it is difficult (perhaps impossible) to obtain the right of use of millions of documents coming from various sources (the authors of the WaCky project wrote on their website “If you want your webpage to be removed from our corpora, please contact us”). So the problem of the copyright on texts used for building corpora is real and capital, but until now a cogent solution hasn’t been found. Maybe alternative ways on the managing of the copyright laws can be found: the traditional copyright, even if it's the most widespread method of protection of intellectual property, is not the only one, and some alternatives emerged in a forefront sector, computer science: in the last few decades, the idea of copyrighted software has been emerging and in the meantime the idea of free software has also emerged, with the birth of various legal models with the aim of protecting and distributing material (first, software, but after also texts of various kinds) but at the same time reserving a few rights, like the paternity on the work, and leaving the other rights for the community, like the right to re-distribute the work without getting the permission of the author each time. The first were licensed by the GNU project, like the Free Documentation License or the Lesser General Public License[7], but the most important for us are the Creative Commons (CC) licenses, established in 2001 and including 6 standard models:

1 Attribution

2 Attribution – non commercial

3 Attribution – non derived works

4 Attribution – share alike

5 Attribution – non commercial – non derived works

6 Attribution – non commercial – share alike

At `www.creativecommons.org` there are all the instruments and troubleshooter guides for a correct use of these licenses, and every model is available in three formats: the *commons deed*, a little resume of the license, the *legal code*, the entire text of the license in legal language, and the *digital code*, a HTML-coded version of the license that can be inserted in the web page with the content that we want to release with this license. The last point proves that CC licenses are widespread among web users, and all around the world and, thanks to the iCommons project, it is adapted to every local copyright legislation where some interest was shown.

For this reason there are a lot of documents released under CC licenses on the Internet, in many languages, and they could be used for

[5] For example limiting the use of limited pieces (500 types) taken from the corpora (ELAN and TRACTOR).

[6] The worst example is the Open Language Archives Community, with this disclaimer: “Open does not mean that users are free to do whatever they like with the metadata, nor does it mean that described language resources are openly available”.

[7] `http://www.fsf.org/licensing/`

building corpora that, thanks to the flexible licenses of their documents, will be easier to use, distribute and manage for every purpose without the usual copyright troubles, and in total respect of the law; in particular licenses like “Attribution” or “Attribution-share alike”, without the “non derived works” or “non commercial” option, are the best for this kind of purpose. So to verify this possibility of building free linguistic corpora from the web we decided to create two general-purpose corpora: the first being a normal corpus of the Italian language from the web, the second made only with Italian-language pages released under one of the CC licenses; afterwards they will be compared in order to understand if CC licenses are widespread enough to use the released documents to build wider linguistic corpora out of the web.

3 Realization of *CopyCorpus* and *CreativeCorpus*

We are now going to create two corpora from the web: *CopyCorpus*, a normal corpus whose documents aren't selected considering the procedure of issue (so these documents could be released with the normal copyright – above all – but also with all the rest of possibilities existent), and *CreativeCorpus*, with documents released exclusively with a CC license. We wanted to create a corpus with only CC-licensed texts and evaluate it comparing with an already existing Italian corpus like *itWaC*, but for an optimal comparison we wished to use exactly the same procedure as the one used with the non-CC web corpus (obviously apart of the selection of CC documents), and this is almost impossible. We decided also not to choose one of the licenses in particular because our purpose isn't to build a large corpora of CC documents directly: the idea is good, but we have to verify its plausibility before proceeding onto a larger – and more defined – project of this kind. We have therefore created two smaller corpora, around 20 million words each, in order to make an easier comparison (the choice of the Italian language is also the best for our linguistic competence in order to be able to make a correct evaluation of the results).

As said in the first chapter of this article, the toolkit we used is BootCaT, and these are the steps of the procedure we followed to build our two corpora:

- Choose seeds/keywords and building n-tuples
- Use n-tuples on Yahoo! and retrieve urls
- Fetch corresponding pages, data cleaning and build corpus
- POS-tagging
- Indexing with CWB

First of all, BootCaT begins with a list of words of our choice to build a corpus which is right for our purposes; in this case we want to build two general-purpose corpora, so we need a list of the most common words in Italian: these words help us to find documents coming from more possible miscellaneous web pages, and get well-balanced corpora. For this step we decided to use the same beginning seed list used for building *itWaC*, the general-purpose Italian corpus of the WaCky project. This list, as explained in Baroni and Ueyama (2006), was built starting with a series of random combinations of frequent Italian words taken from the basic vocabulary and from the *La Repubblica* corpus, later organized in tuples forming a list of 1000 lines.

flotta 'fleet' coppa 'cup'
procuratorio 'attorney' assicurativo 'insurance'
parallelo 'parallel' bandito 'bandit'
direttiva 'directive' commettere 'commit'
polizza 'policy' polemico 'controversial'
statua 'statue' atletica 'athletics'
abilità 'ability' costoso 'expensive'
gente people' celebre 'famous'
minuto 'minute' coordinatore 'coordinator'
torto 'wrong' suggerimento 'hint'
quattro 'four' medesimo 'same'
dimostrazione 'demonstration' spedire 'send'
nastro 'tape' occupazionale 'occupational'
pacco 'pack' concentrare 'concentrate'
preoccupante 'alarming' anima 'soul'
puntuale 'punctual' allargamento 'enlargement'
alleato 'ally' osservatorio 'observatory'
astensione 'abstention' maresciallo 'marshal'

smentita 'denial' mese 'month'
consapevolezza 'consciusness' mobilità 'mobility'

Table 1. First 20 lines from the seed list

Every line of this list is then used to make automated searches on Yahoo!'s search engine with a dedicated tool (that works via API)[8], retrieving url of sites containing documents with these combinations of words: the default configuration of this tool is 10 results for tuple, and we left it so. We did this twice: the first with a simple search specifying only the language, and the second one where we selected, thanks to a Yahoo! option, only documents released under CC licenses. There's no filter on the results ("in the name of the modularity", as said by the author of the script[9]), so if we want to remove duplicates and meta-information we can use a simple Unix command that does this. This is only the preliminary stage of data cleaning, but it will then be done again, in particular with the content of this urls.

Now we have two lists of urls, and we have to download the content of each one. The script written for this work downloads pages "applying a heuristic method to look for the 'content-rich' section of a page, and removing the rest"[10], also ignoring all the urls that don't begin with *http* and finish with non-html formats like *.doc*, *.jpg*, *.pdf*, *.ppt* ecc. The tool also offers the possibility to improve the downloading by discarding linguistically non-interesting material (boilerplate) using lists of "good words" (that must be in our documents) and/or "bad words" (that must not be in our documents). The use of these lists could be very favorable, especially in the case of the construction of a general-purpose corpora or when we have to work on a well defined field. We applied the first option, by using a list made with the most 100 frequent words of *itWaC*, and in this way getting the content of our two corpora. Then we deleted from the corpora other linguistically non-interesting elements, like sections containing special encoding or documents with identical content, with suitable scripts still contained in BootCaT.

At this point the building of the two corpora is finished. The next steps have the aim of exploring the corpora in greater depth; the first is the Part-Of-Speech tagging, where there could be some difficulties, depending on the internal configuration of the tagger that could make mistakes in recognizing the right POS or lemma of a token. However, this will be another interesting aspect of the corpora building that we'll see in the next chapter, with concrete examples.

The last stage of this work is indexing, a very important operation that permits advanced exploring options. The best tool for this is the IMS Corpus WorkBench (CWB)[11], that completes the indexing of our corpora and makes it ready for exploration with the Corpus Query Processor (CQP), a search program associated to the CWB. In this step we have only to specify the language of our corpora, choose their names and write a short description.
Now we have our two corpora, *CopyCorpus* and *CreativeCorpus*; the first is made up of 16.592.419 words and its size is 137 megabyte; the second 18.293.242 words and 148 megabyte. So we succeeded in getting two corpora with similar dimensions, created following the same procedure apart from the limitation to CC documents for CreativeCorpus. The next job is an analysis of their contents, that will help to understand how far it's possible to create a corpus only with CC-licensed documents.

4 Analysis and comparison between the two corpora

We now have to choose a good method of evaluation and comparison between our two corpora. The best thing to do is to consider previous experiences that may be related to this work. Using BootCaT, Sharoff (2006) shows us the realization, of three corpora of English, German and Russian (*I-EN*, *I-DE*, *I-RU*), and he analyzes them on the basis of their composition - classification of corpora samples into categories like Authorship (*single*, *multiple*...), Mode (*written*,

[8] We use Yahoo! because Google doesn't release API keys anymore; moreover Yahoo! is the best choice for us thanks to its search engine that fully implements searches on CC documents. For more informations: `http://developer.yahoo.com/search/web/V1/webSearch.html`

[9] From the documentation of `collects_urls_from_yahoo.pl`, contained in `http://sslmit.unibo.it/~baroni/new-boot.tar.gz`

[10] From the documentation of `retrieve_and_clean_pages_from_url_list.pl`, contained in `http://sslmit.unibo.it/~baroni/new-boot.tar.gz`

[11] `www.ims.uni-stuttgart.de/projekte/Corpus-Workbench`

spoken...), Domain (*politics*, *life*, *arts...*) etc. - and with frequency lists, using the log-likelihood as association measure. Another work we've considered is Ueyama and Baroni (2005), a little diachronic study on Internet Japanese, where an analysis is made which is very similar to the one by Sharoff (2006) but, as far as the analysis by categorization is concerned, they distinguish between **topic domains** (*natsci*, *socsci*, *business*, *life*, *arts* etc.) and **genre types** (*blog*, *BBS*, *argessay*, *commerinfo*, *teaching*, *news*, *magazine*, *review* etc.). On the contrary Ferraresi (2007) doesn't use an analysis by categories, but he worked out a more detailed method of comparison among frequency lists for comparing his *ukWaC* with *BNC*: like Sharoff (2006) he uses the log-likelihood but Ferraresi creates separate lists based on the main grammatical categories in English: nouns, verbs, adjectives, adverbs ending in *-ly* and function words.

On the basis of all this, we decided to proceed with a double analysis: a categorization like Ueyama and Baroni (2005) and a comparison among frequency lists like Ferraresi (2007), but slightly changed to make these analyses more suitable for our two corpora.

4.1 Topic domains and Genre types

As previously stated, for this categorization we used the same topic domains and genre types used by Ueyama and Baroni (2005), with little changes, by substituting old categories or by introducing new ones: the changes concern only genre types, and this is due to the great growth in recent years of the new Web 2.0 applications, and we wanted to insert them in this classification (for example with the introduction of *social networking*, *video-pics gallery* and *wiki*) or, if they were already present, give them more importance, like *blog*, that nowadays entirely includes the old category *diary*. Then we chose 100 random documents (with automated tasks in order to assure a more random possible selection) from *CopyCorpus*, and another from *CreativeCorpus* and manually classified each one on the basis of the topic domain treated in it and on the genre type used to write it. These are the results:

TOPIC DOMAIN	**Copy Corpus**	**Creative Corpus**
appsci	15	9
arts	3	8
business	9	1
error	2	7
leisure	26	32
life	10	7
natsci	3	6
sosci	32	30
TOTAL	100	100

Table 2. Topic domains

GENRE TYPE	**Copy Corpus**	**Creative Corpus**
argessay	9	10
blog	21	50
commerinfo	5	-
error	7	8
essay	1	4
faq	1	-
forum	7	-
groups	2	-
info	9	9
instinfo	7	2
magazine	1	-
news	9	9
personal	8	2
report	3	3
review	2	2
social networking	2	-
speech	1	-
teaching	1	1
video-pics gallery	2	1
wiki	2	1
TOTAL	100	100

Table 3. Genre Types

We can see a substantial likeness in topic domains: the most important differences are in categories and for example it was foreseeable that we could find only a few in *CreativeCorpus* like *business*. In general we can see that the most representative categories are the same in both the corpora, and the disparity increases going down to the less common domains: *CopyCorpus* has more *appsci* and *life* than the other, but CreativeCorpus has more *arts*, *natsci* and *errors*[12]. We can say that *arts* could be more represented in *CreativeCorpus* because of the fact that Creative Commons itself offers adequate instruments to license creativity works, but it's advisable also to consider the second table to obtain a better-founded analysis.

The results for genre types are remarkable: here we have more unbalanced results, with *CreativeCorpus* lacking a lot of categories, and the remaining sharing a lot of the pages. But the most important result is that the half of the samples considered out of *CreativeCorpus* are blogs. This isn't completely unexpected, but the data confirmed it, and we found an explanation of this in the fact that the use of CC licenses among bloggers (in this case in Italian pages, but probably it's true also for other languages) is very widespread, because they can be considered an excellent instrument for licensing this kind of textual contents on the web; moreover, the blog shape turned out as an extremely modular means of communication and suitable for very varied purposes, and for these reasons it was adopted by a large variety of web users[13]. So the combination “blog+Creative Commons license” was shown to be not only very popular also among common users, who nowadays use a great quantity of Web 2.0 applications (not only blogs, but also photo and videosharing hostings, content management systems and so on), but also – and for this reason – very useful to build corpora from the web.

At this point, we can say, based on this first part of the analysis, that the sites licensed under CC licenses deal, in general, with the same subjects of the majority of Italian sites considered in its totality, especially free times and subjects of social interest, and much of the difference between the two corpora concerns the genre containing these domains. So we can say that, talking about contents, there is a substantial similarity of contents; but we need the next analysis to make this more clear.

4.2 Word list comparisons

We think that, even if the previous analysis has given us an idea of the composition of our corpora, it is not clear enough to be sufficiently well-informed about the content of the two corpora. We therefore need a more detailed analysis, and this time the construction of word lists is necessary for further comparison; like Ferraresi (2007), we decided to create separate lists based on the principal grammatical categories, so we built several word lists for *CopyCorpus* and *CreativeCorpus*, each contaning the word items that were identified by the tagger as belonging to the main Italian part-of-speech categories: nouns, verbs and adjectives (this procedure obviously relies heavily on the tagger's performances, but we'll see very interesting results also coming from the tagger's errors). Each list has been then compared with its counterpart via the log-likelihood association measure, taking *CopyCorpus* as a benchmark when calculating the key words of *CreativeCorpus* and vice versa, and then sorting the results according to their score, from the highest to the lowest (and selecting only the first 50 elements of each list; for every element we then analyzed 50 random concordances from the corpora). This procedure made it possible to obtain a crossed analysis that can show us the features of our two corpora considered together, because these frequency lists give relatively typical words of one corpus only when compared to the other, and not absolutely; the log-likelihood has been chosen as the best association measure because of this possibility to consider together (and comparing) the content of our two corpora. So this analysis shows us the differences between *CopyCorpus* and *CreativeCorpus*, but we could understand how the two corpora are similar considering the differences that did not emerge from this analysis.

Lemma	**Number of occurrences in *CopyCorpus***	**Total number of occurrences**	**Log-likelihood ratio**
amore 'love'	5637	8247	1396.6
message	1139	1189	1323.4
send	904	936	1090.5
cuore 'heart'	4497	6716	983.7

[12] *errors* contains various kind of useless pages because their contents are, for example, bad machine-generated texts or other linguistically uncorrected material.

[13] Some blog hosting services directly offer the opportunity to choose by default a CC license (like Motime-Splinder), but this option is equally chosen by users that use other blog services that don't include them as an internal option (Blogger, Wordpress etc.)

clic	875	959	825.3
missione 'mission'	2265	3087	823.2
chiesa 'church'	5699	9075	804.4
comma	3252	4822	743.1
etichetta 'tag'	1574	2043	723.4
uomo 'man'	13092	23165	678.0

Table 4. Example of frequency list. First 10 most typical nouns of *CopyCorpus*

Lemma	**Number of occourrences in *CreativeCorpus***	**Total number of occourrences**	**Log-likelihood ratio**
punto 'point'	37340	47435	14630.5
commento 'comment'	21462	28350	6762.7
link	9089	11333	3904.7
gen 'jan'	2561	2584	3119.7
mail	4240	4739	3087.7
dibattito 'discussion'	4776	5500	3024.9
forum	4535	5360	2538.4
nov	2144	2193	2406.0
moto 'motorbike'	4103	4847	2301.9
dic 'dec'	1925	1956	2244.1

Table 5. Example of frequency list. First 10 most typical nouns of *CreativeCorpus*

About the most frequent nouns of *CopyCorpus*, we have a huge quantity of religious terms (in particular Christian-Catholic): they could be fragments coming from the Bible and texts of various kind (articles, essays ecc.) of religious argument; other very common words not exclusively belonging to this domain are well represented in these religious texts: *amore* 'love' 36%, *uomo* 'man' 40%. Another well-represented domain in *CopyCorpus* concerns nouns belonging to the computer sphere, in particular Internet's lexicon (*clic*, *server*, *font*) including common words now belonging to the language of the web (*send*, *message*, *etichetta* 'tag'), and bureaucratic language, with words like *comma*, *tabella* 'table', *membro* 'member', *numero* 'number'. Other common words like *occhio* 'eye', *oggetto* 'object', *albero* 'tree', *fuoco* 'fire', *corpo* 'body' cannot be classified as belonging to a domain in particular, but they are associated with a lot of fiction (with a consistent quantity of fanfiction), and a lot of machine-generated text made by the various automated-translation services provided by the most famous portals.

Also in *CreativeCorpus* we have a lot of computer terms, but in this case we have almost all nouns belonging to the page structures (templates and CMS): for example, we can often find *nickname* in the following situations, alternatively: *effettua il login per riservare il tuo <nickname>* 'log in to reserve your <nickname>', *registrati per riservare il tuo <nickname>* 'register to reserve your <nickname>'. For the rest there's a lot of words belonging to the motor world (*vettura* 'car', *berlina* 'sedan', *km*, *cv*). Remaining words (like *euro*, *mercato* 'market', *polizia* 'police', *stampa* 'press', *carcere* 'jail') are of various kinds, but a great quantity of them come from pages containing articles (most of the time in the form of blog posts) talking about news of great topicality or journalism.

Very similar results were found with verbs and adjectives: *CopyCorpus* seems to include a larger variety of documents but with some topics emerging in particular: religion and others like (fan)fiction, videogame guides, words coming from professional spheres and a large quantity of badly-made automated-translated text. *CreativeCorpus* instead contains a lot of words that we can consider boilerplate coming from CMS' page structures or articles on topical subjects, or journalistic material as well as discussions on comments coming from blog posts and, talking about very particular spheres, articles on the motor world. However the two corpora are similar talking about their content considered in general: these differences concern well-defined domains and there aren't emerged stronger (and so due considerations about it) differences that showed us a strong imbalance on too much particular themes or domains.

5 Conclusion

To sum up, considering these results we can say that even if CC-licensed documents' contents and genres are circumscribed, CC licenses are widespread enough to build a balanced corpus, both in Italian and in general-purpose, with wide variety, in particular when talking about blogs. As we just told, there are some differences between the two corpora that emerged from the last analysis, but as we said describing the purposes of the word list comparisons, considering also the differences that did not emerge from the analysis, we can say that there aren't strong differences that let us saying the two corpora are too much different about their being balanced (without a consistent unbalance in favor of one (or more) particular sector, genre or topic) and general-purpose (that cover most possible genres and topics, theoretically useful for every linguistic purpose) corpora. We have also a few advantages when using only CC-licensed web pages; in particular, we can be almost sure that there is a human creator behind them, unlike the great quantity of machine-generated articles (full of errors) we found in the normal corpus; and, considering that the greatest part of linguistically non-interesting material comes from the structure of the pages (many of them made with Web 2.0 applications), this undesired textual material is easier to eliminate than other kinds of boilerplate.

As we said, now this test might have further developments, such as, for example, how (and how much) CCs are used in other language web pages, and also see how their application changes in building web corpora.

References

Adriano Allora and Manuel Barbera. 2007. *Il problema legale dei corpora. Prime approssimazioni*. In M. Barbera, E. Corino, and C. Onesti. 2007. *Corpora e linguistica in rete*. Guerra edizioni, Perugia. 113.

Marco Baroni and Silvia Bernardini. 2004. *BootCaT: Bootstrapping corpora and terms from the web*. *Proceedings of LREC 2004*, 1313-1316.

Marco Baroni and Motoko Ueyama. 2006. *Building general- and special purpose corpora by Web crawling*. *Proceedings of the 13th NIJL International Symposium*, 31-40.

Silvia Bernardini, Marco Baroni and Stefan Evert. 2006. *A WaCky introduction*. In M. Baroni and S. Bernardini (eds.). *WaCky! Working Papers on the Web as Corpus*. GEDIT Edizioni. Bologna. 9-40.

Adriano Ferraresi. 2007. *Building a very large corpus of English obtained by web crawling: ukWaC*. Graduation thesis. Università di Bologna.

Adam Kilgarriff and Gregory Grefenstette. 2003. *Introduction to the special issue on the Web as corpus*. *Computational Linguistics*. 29(3).

Steve Lawrence and C. Lee Giles. 1999. *Accessibility of Information in the Web*. *Nature*, 400. 107-109.

Anke Lüdeling, Stefan Evert and Marco Baroni. 2007. *Using Web data for linguistic purposes*. In M. Hundt, N. Nesselhauf and B. Caroline (eds.). *Corpus linguistics and the Web*. Rodopi. Amsterdam. 7-24.

Serge Sharoff. 2006. *Creating General-Purpose Corpora Using Automated Search Engine Queries*. In M. Baroni, and S. Bernardini (eds.). *WaCky! Working Papers on the Web as Corpus*. GEDIT Edizioni. Bologna. 63-98.

Motoko Ueyama and Marco Baroni. 2005. *Automated construction and evaluation of Japanese Web-based reference corpora*. In *Proceedings of Corpus Linguistics 2005*.

Stefano Vegnaduzzo. 2007. *Scoperta automatica di relazioni lessicali usando il world wide web*. In R. Maschi, N. Penello and P. Rizzolatti, *Miscellanea di studi linguistici offerti a Laura Vanelli da amici e allievi padovani*. Forum Editrice. Udine.

Using e-data for the study of language change: a comparative study of the grammaticalized uses of French *genre* in teenage and adult forum data

Emeline Doyen
University of Leuven
emeline.doyen@student@kuleuven.be

Kristin Davidse
University of Leuven
kristin.davidse@arts.kuleuven.be

Abstract

In this paper we investigate the layering of lexical and grammaticalized uses of the French noun *genre* in Internet data from teenage and adult forums. Qualitative and quantitative analysis of these two datasets confirms the hypothesis that the process of grammaticalization is more advanced in the teenage data. More specifically, these data contain many more uses of *genre* in which it has detached itself from its source structure, the noun phrase, viz. the quotative uses indexically associated with teenage language as well as qualifying particle and discourse particle uses. We conclude that, while informal, spoken language is generally recognized to be the primary locus of language change and innovation, more attention should be paid to the special role played by teenage language in these processes. As corpora of teenage language are not generally available, Internet data from forums or communities targeting specific age groups are an important resource for carrying out research into ongoing language change.

1 Introduction[1]

The layering of lexical and grammaticalized uses of type nouns has recently been studied in a number of languages such as English (Aijmer 2002, Denison 2002, De Smedt, Brems & Davidse 2007), German (Diewald 2006) and French (Fleischman & Yaguello 1999). In discussions comparing the degree of grammaticalization in these languages, we have heard the opinion ventured that English type nouns such as *sort* and *kind* have grammaticalized more than equivalent type nouns in French.

The source structure of the grammaticalized uses of English *sort* and *kind* is the binominal construction (Denison 2002: 2), which describes a subclass of some superordinate category, as in

(1) Pountney sees the opera as a study of two kinds of woman, the fleshly and the spiritual. (CB – Times)

The type noun is the head of this construction and is used in its main lexical sense, referring to a class of things.

The binominal construction was first reanalysed into constructions that remained *within* the confines of NP structure, and in which the type noun contributed to nominal grammatical functions being expressed, as in (2) and (3), in which *sort of* is enclitic to the determiner.

(2) We were only able to respond that we were unaware of any evidence linking plastic milk bottles and cancer ...] Unfortunately these sort of scare tactics do a lot of harm. (CB-Oznews)

(3) One of them called optimistically for the enshrining of the World Cup triumph last June as some kind of national treasure. (CB – Times)

The construction illustrated by (3), in which *some sort of* ‘qualifies’ – in this case ironizes - the nominal description that follows it, then became the source structure of a new chain of reanalyses. According to Denison‘s (2002) recon-

[1] Our sincere thanks go to the three anonymous referees for their very generous and incisive comments which helped us improve the first version of this article.

struction of these changes, the qualifying force of *sort/kind of* first extended its scope to other classes than nouns, such as verbs (4), adjectives (5) and even whole utterances (6). These extended qualifying uses then semantically bleached into discourse particles which no longer have clear scopal domains (7). The most recent development is the emergence of onomatopoeic and quotative uses (Aijmer 2002: 168) (8-9).

(4) and they kind of group – put people into kind of categories (qtd Denison 2002: 12)

(5) Then later in the night we took a walk in our underwear around the campus. That was sorta weird. (www.yaledailynews.com/article.asp?AID=23414)

(6) CMG: Sort of like In The Fishtank? -- Beam: Kinda. (www.cokemachineglow.com/feature/interview/beam.html)

(7) ^well I !don't think .^it's ^((sort of a)) . a com:plete con:cl\usion= you're sort of ^left with the - - you ^sort of [∂:m] – it's ^sort [∂?] an :end to a :story in a :w\/ay= . (qtd Aijmer 2002: 189)

(8) I've ^neve s/\een a 'sortof# ^bottle 'after :b\ottle# . sort of ^pop 'pop p/opping# âll the t/\ime# (qtd Aijmer 2002: 186)

(9) im just being kinda hey i can hear murkin (qtd De Smedt et al 2007: 248.)

Importantly, these new grammaticalized uses have syntactically detached themselves from the nominal source structure. These NP-external uses are advanced and innovative forms of grammaticalization. They are not directly predictable from the typical decategorialization cline assumed for nouns, which remains within NP-structure and whose end stages are envisaged as clitics or affixes (Hopper & Traugott 2003: 110).

To compare the degree of grammaticalization and innovativeness, Willemse, Brems & Davidse (2007) compared all the uses of English *sort* and *kind* with French *sorte* and *espèce*. For English they looked at data from the formal written Times subcorpus of COBUILD and the informal spoken COLT corpus, The Bergen Corpus Of London Teenage Language. For French, data were examined from the formal written *Frantext* corpus and the spoken LANCOM corpus. This study confirmed the expectation that French *sorte* and *espèce* do not have grammaticalized uses external to NP structure. However, in Fleischman & Yaguello (2004) it is suggested that it is the French type noun *genre* that is developing such uses. In this article we want to investigate in a systematic way if French *genre* has developed all the NP-external uses manifested by *sort* and *kind*.

The view that casual spoken exchanges between peers constitute the most important locus of language change (e.g. Halliday 1978) has come to be generally accepted. In this respect, it is a pity that the main dataset of spoken French, the Lancom corpus, is rather restricted and contains too few instances of the type noun *genre* to form a solid basis for our investigation. We will, therefore, examine the grammaticalization of *genre* in Internet forum data, an informal register closer to spoken language than any other accessible sources available at the moment. We will investigate whether *genre* is found in all the grammaticalized construction types established by De Smedt, Brems & Davidse (2007) for English type nouns, including those extending beyond NP-structure. In addition, our hypothesis is that the process of grammaticalization of *genre* is most advanced in young people's informal language. Some recent studies have shown that particularly discourse features may change within a brief period in young speaker's language. This was established for quotatives by, amongst others, Golato (2000), Macaulay (2001), and Tagliamonte & D'Arcy (2004) and for intensifiers by Ito & Tagliamonte (2003) and Macaulay (2006). Two of the innovative uses of *genre* that we are interested in are in fact intensifiers and quotatives. It was the strong intuition of the first author of this paper, a young native speaker of French, that *genre* offers a clear example of innovative uses being typically associated with teenage speech. Therefore, we will compare the uses of *genre* in data from teenage and adult forums. The analysis of these data will be both qualitative and quantitative.

The structure of the paper will be as follows. After discussing the data in section 2, we will, in section 3, describe the different construction types that *genre* occurs in, noting the structural features and semantic-pragmatic values observed in the data. In section 4, we will look at the quantitative distribution of these construction types over the teenage and adult data.

2 Data

The great advantage of Internet data for a study of ongoing language change is that they give access to recent innovative usage. The forums our data were culled from are *Adojeunz.com* (http://www.adojeunz.com/forum/index.php) and *Discutons.org* (http://www.discutons.org/Debats_ generaux_d_actualite-Forum-3.html). Both forums were carefully chosen to optimally represent the target populations. *Adojeunz.com* is used by teenagers between 12 and 20 years old. *Discutons.org* may be used by a larger public but sections were chosen that were likely to be written and read by adults, viz. politics and current affairs.

The main disadvantage is that, unlike compiled corpora, the Internet is not a finite database of fixed size. Therefore, there is no easy way of relating attestations to the overall size of the corpus or of subcorpora. Yet, for this study comparing the grammaticalized uses of *genre* in teenage and adult language, one would like to know how common these uses are, for instance, per 100,000 words in the respective corpora, i.e., what their normalized frequencies are.

To get at least an idea of the normalized frequencies of the various uses of *genre*, the first author of this study manually compiled pilot corpora of approximately 120,000 words from both forums, PC-T (pilot corpus teenage data) and PC-A (pilot corpus adult data). As she had to copy-paste every post, the whole procedure was very time-consuming, but it was the only way to relate occurrences to overall size. While the language of forum debates is inherently informal and dialogic, consisting of question-answer pairs, and statements reacting to previous statements (Martin 1992), some differences between the two forums should be noted. The teenage forum is more informal than the adult forum, which is reflected in the topics of the exchanges, e.g. posts about singers and actors versus posts about politics and current affairs. There was also a difference in the number of post compiled for PC-A and PC-T because adults' posts, which build up argumentations, tend to be much longer than teenagers' posts, which exchange evaluations and comments. The PC-T (120,857 words) contained 100 instances of *genre*, while the PC-A (119,381 words) featured only 51 instances of *genre*. Conclusions to be drawn from the normalized frequencies of the distinct uses within these two corpora will be discussed in section 4 below.

To arrive at datasets sufficiently large to describe the grammaticalized uses of *genre*, Doyen complemented the tokens in the pilot corpora with examples culled from the two forums with the Google search engine. Samples were thus arrived at of 650 tokens of *genre* for both the teenage (T) and adult (A) data. Accessing the forums on the same days, she collected 250 tokens of *genre* for T and A on 6.12.2008 and 7.12.2008, and 300 additional ones for T and 349 extra ones for A on 12.7.2009 and 13.7.2009. Thus, samples of 650 tokens could be put together for the two target populations. These samples yielded 482 relevant tokens for T and 484 relevant tokens for A. Relevant tokens are the grammaticalized uses and their source construction, binominal NPs with head *genre*. Examples irrelevant to this study include expressions in which *genre* functions as postmodifier, e.g. *une guerre d'un genre nouveau*, and fixed expressions such as *être son genre*. We considered these datasets as random samples of the various uses, whose distribution in terms of proportions of the teenage and adult datasets will be discussed in section 4. The description of the uses in section 3 is based mainly on these datasets. However, as is well-known, even reasonably extended datasets do not contain instances of all possible variations of constructions. Therefore, we also refer to examples from the literature and other Internet sites.

3 Construction types with *genre*

3.1 *Genre* as head noun of a binominal NP

The binominal construction with *genre* as head is the source construction of the grammaticalized uses that will be discussed in sections 3.2-3.5. In it, we find the lexically full use of *genre* referring to a subtype followed by *de* + a second noun (henceforth N2) designating a superordinate class, e.g.

(10) Ce genre de musique pour étudier, c'est la classe. (T)

(11) Ca peut paraître étrange mais mes tattoos, aussi petits soient-ils pour l'instant, m'ont fait oublier la plupart de mes défauts physiques, je ne sais pas, peut-être parce que, en quelque sorte, le corps devient un genre nouveau d'oeuvre d'art, [...] (T)

As *genre* in this construction still has its full lexical weight, it can be descriptively modified by an adjective, such as *nouveau* in (11). Such binominal NPs always realize generic reference referring to the whole subclass and are intrinsically concerned with generic and taxonomic interpretations of the world. Their structure can be represented as follows, with optional elements put between brackets:

determiner + *genre*/head (+ adjectival modifier) + *de* + N2 (+ adjectival modifier)

3.2 *Genre* as postdeterminer

The first grammaticalized use of *genre* that we discuss is the postdeterminer construction, in which *genre* occurs after the determiner and supplies additional determining (i.e. grammatical) information. In this construction, *genre* has been demoted from head status, which is shown by the fact that it is always singular, even when the determiner is plural, as in (13). This indicates decategorialization, i.e. loss of normal morphosyntactic behaviour (Hopper & Traugott 2003) of the noun *genre*.

(12) le problème du rejet que produit des phrases comme "ce sont surtout des maghrébins qu'on voient mettre le boxon" vient beaucoup du fait que ce genre de phrase sert justement à masquer toute l'autre délinquence qui est non-dite (A)

(13) Après, on se scandalise que le créationnisme gagne du terrain, mais quand on fait passer ces genre de conneries pour des prédictions valables, [...] (www.comlive.net/ Honte-A-Tf1-Honte-A-Mary line)

In these examples we can also observe a semantic shift, as *genre* no longer refers to a subclass that is part of the structure of the world, but functions within a unit realizing the textual relation of anaphora. This meaning can also be expressed by determiner plus anaphoric *tel* (Van Peteghem 1995), which can replace *ce(s) genre de*: *une telle phrase, de telles conneries***.** Determiner and *genre de* form a complex determiner which singles out referents in terms of contextually presupposed defining qualities. For instance, in (13) *créationnisme* evokes qualities such as ‘non-scientific’. It is to these implied qualities that *ces genre de* points back and in this way sets up a contextual generalization which refers to *créationnisme* as well as to other such theories. Structurally, determiner + *genre de* form one unit which realizes the general determining function in the NP. Within this unit a second postdeterminer can occur, such as *même* in (14).

(14) Pour ce qui est de la couleur de peau, ça relève un peu du même genre de phénomène. (A)

The second noun is the head of the NP, and can take descriptive modifiers, as in

(15) C'est toléré ici, ce genre de propagande réactionnaire? (A)

The overall structure can thus be represented as follows:

complex determiner [determiner + (adjective/postdeterminer) + *genre de*/postdeterminer]+ N2/head (+ adjective/descriptive modifier)

The instructions given by complex determiners with *genre* for the contextual retrieval of defining qualities of the referent(s) can not only be anaphoric but also cataphoric, and may even not involve an antecedent or postcedent in the strict sense at all but a more general cohesive relation. These three subtypes manifest very clear preferences as to determiners collocating with *genre*. Anaphoric relations are typically expressed by demonstrative determiner *ce* + *genre*, as in (12) and (13), but are occasionally also construed by *le même genre* (as in (14) above). Cataphoric relations, which point forward to defining characteristics expressed by relative clauses or other postmodifiers, are mostly expressed by the definite article *le* (occasionally supported by *même)* + *genre*, as in (16), but we also find some possessive determiners + *genre*, as in (17).

(16) Il avait tout à fait le même genre de question que dans la vidéo de Desaix. (A)

(17) Puis ton genre de discour [sic] en faveur de Jeanot par ex ... (A)

More general textual relations are expressed by the interrogative determiner *quel* + *genre* (18), or by quantifiers such as *aucun* + *genre* (19). Even though no antecedent is referred to, these determiner complexes still invoke a relation to contextually inferable qualities. Interrogative *quel* has a basic sense of “variable”, for which the answer has to provide a specific value. In an example like (18), the value corresponding to *quel genre de parents* is contextually implied to involve negative qualities.

(18) Avant de condamner les enfants, faudrait peut etre savoir quel genre de parents ils ont. (A)

Quantifiers like *aucun* tend to be used in contexts with counterexpectations (McGregor 1997: 281-2). For instance, *aucun genre de répression* in (19) denies the expectation of *répression* set up in the preceding clause (about Brazilian peasant women) for Brazilian enterprises.

(19) Contrairement à ce qui est arrivé aux femmes paysannes, les entreprises n’ont dû supporter aucun genre de répression pour parvenir à

leurs fins. (http://www.genreenaction.net/spip.php?page=imprimer&id_article=6489)

3.3 *Genre* as part of nominal qualifying construction

A second reanalysis, and grammaticalization, of the binominal construction is the nominal qualifying construction. It is commonly accepted that this reanalysis is enabled by bleaching of the lexical ‘subtype’ meaning into the pragmatic sense of ‘peripheral membership’ (e.g. Denison 2002). A symptom of *genre*’s demotion from head status in this construction is the tendency of the gender of the determiner to be governed by N2, rather than by *genre*. This is illustrated by (20), in which *une* agrees with (feminine) *pétition,* not (masculine) *genre*.

(20) J'y vois une genre de belle pétition vidéo sur l'état du monde. (citizen.nfb.ca/node/23901&term_tid=54 - 76k)

As we are dealing with Internet data, the question might be raised whether (20) is not simply a grammatical mistake. We therefore explored this phenomenon in more detail by searching the Internet with Google for random combinations of *genre* followed by a feminine noun that could be expected to typically trigger qualifying uses. We noted the number of occurrences with both feminine and masculine determiner, and found that the feminine form often predominated, e.g. *une genre de suite* / *un genre de suite*: 493 – 70; *une genre de thérapie* / *un genre de thérapie*: 208 – 142; *une genre de réplique* / *un genre de réplique*: 116 – 10. This argues for the view that we are dealing with a motivated pattern of change, not mistakes, here. In addition, it can be noted that gender agreement with N2 is well-established in non-Internet examples of qualifying uses with other type nouns, particularly pejorative qualifying uses with *espèce*, such as *un espèce de crétin,* an example found in the literary and academic corpus *Frantext*, in which *un* agrees with masculine *crétin*, not feminine *espèce*. The fact that the type noun no longer determines the gender marking is a sign of its decategorialization, accompanying its grammaticalization, or shift towards expressing grammatical meaning.

In nominal qualifying constructions, the nominal classification expressed by N2 is qualified, i.e. hedged (21), softened (22), pejorized (23), or otherwise nuanced. All these uses are based on the notion that the classification is only approximate.

(21) Et puis j'ai fait un genre de malaise en amphi (dans un coin au fond comme une mauvaise élève). (T)

(22) Un genre de masturbation mentale collective "ouais Adojeunz ça pue ici c'est mieux d'abord" (T)

(23) ... n'est pas assuré ou alors trés mal contre les incendies [...] Hors ce n'est que ce genre d'épaves roulantes qui sont incendiés !! (A)

3.4 *Genre* as qualifying particle

In this use *genre* is no longer part of the structure of the NP, but has broken free, as it were, of its nominal dependency structure. A formal reflex of this is that it is no longer followed by the particle *de*. *Genre* as such is used as a particle that can move very freely in phrase and clause structure, able to hold almost any unit or subunit in its scope. As in the nominal qualifying construction just discussed, its meaning is to qualify the description of whatever element it has in its scope as approximate or imperfect in relation to the instances being depicted.

When used as qualifier of an adjective, *genre* indicates that the qualitative description is approximate (24), or it modifies the degree to which the quality is present in the instance (25). Likewise, with verbs, *genre* can function either as approximator or as degree modifier. In (26), it can be rephrased as ‘so to speak’, while in (27) it invokes an assumed norm with reference to which the force of the verb is heightened.

(24) j'ai un percing [sic] TROOOOP bo sur la lèvre genre bleu pis vert (T)

(25) ce putin de chien qui m'accueille avec un superbe pipi sur le carrelage ... Mais le pipi genre normal quoi. (T)

(26) Au début, je voulais genre faire des fiches sur tout mais je suis vite redescendue sur terre! (http://edp.ipbhost.com/lofiversion/index.php/t83358-50.html)

(27) en voyant mes cheveux elle s'est genre exclamée : "en 30 ans de carriere, jai jamais vu ca". (www.madmoizelle.com/forums/forum-coiffure/13794-special-cheveux-epais.html- 53k)

Genre is also used to mark numbers as only approximate, e.g. (28), and it can hold whole NPs in its scope, e.g. (29), in which it marks the examples listed as typical ones. Finally, *genre* can hold prepositional phrases (30) or whole clauses (31) in its scope.

(28) Où est ce que je peux trouver des tapis de souris pas cher ? (Genre 1€) (T)

(29) Ils m'offraient des CD genre Billy Crawford (?) ou encore la schtroumpf party (T)

(30) Ces colonies, c'est un peu comme des villages complets clé en main, installés genre au milieu de zones palestiniennes. (A)

(31) Et sinon, t'as pas essayer de passer un coup de fil à son lycée tout simplement? genre pour le demander au tel un truc comme ca. (A)

Two features found in some qualifying particle uses and which extend the semantic-pragmatic value of the nominal qualifying use are the exemplifying meaning component, as in (29) above, and what could be called a semi-quotative feature, illustrated by (32).

(32) qui alors est devenue un mythe qu'on considère comme fabuleux genre l'atlantide pour nous, ... (A)

The slogan *l'atlantide pour nous* occurs in a structural position where one would normally have expected a NP. In such examples, *genre* grammatically re-categorizes, so to speak, longer utterances so that they can function in structural slots that normally do not take clauses.

A final specific use of qualifying particle *genre* that has to be singled out is its sentential use, illustrated by (33). Here *genre* has scope over a whole proposition, qualifying its truth or accuracy, and often conveying sarcasm or irony (which may be accompanied in speech by facial expressions such as a smile or raising one’s eyebrows).

(33) Oh non il ose tout notre maitre capello! Une forme olympique. Voila quelques mots qui devraient t inspirer. Genre. Haut lin pique les cuisses. Oh non j y arrive pas moi! (A)

3.5 *Genre* as discourse marker

As discourse marker, *genre* is not tied to grammatical class boundaries anymore, and lacks clear indications of scopal domain. These uses of *genre* apply more diffusely to the discourse. They are used as indicators of tentativity, and as fillers and hesitation markers, often co-occurring with other such markers and fillers. They are probably a further development of the ‘approximator’ value of the qualifying use, resulting from semantic bleaching and (inter-)subjectification: they signal speaker attitude as well as speaker attention to the hearer’s face (Traugott & Dasher 2002), generally conveying solidary or non-dominant social values, e.g.

(34) Genre ouais, je savais à peine me lever... Donc pour m'habiller... Hum. (T)

(35) mouais... genre on laisse couler, ça passera... ou pas. bof bof (A)

3.6 *Genre* as quotative marker

Genre can also be used to introduce quoted material. This may be part of direct speech in its traditional sense, where *genre* can function on its own as a quotative marker (36), but is also often used together with *être* (37) and *faire* (38). The quoted material may also be inserted in structural positions where it is less usual, such as post-nominal position in (39). Finally, *genre* may also introduce onomatopoeia (40), rather than quoted utterances in the strict sense.

(36) Mais ma mère AHAH. Genre: "Ouais, comme là maintenant quoi ! Un petit verre dans le nez, et on arrête pas de parler !" (T)

(37) elle était genre, "Oh, mon dieu, c'est mes reins ? (dr-house.xooit.tv/t1843-Interview-de-Alloy.htm)

(38) Rha pis jme rapelle du gars qui chantait en italien et qui faisait genre c'moi le chef d'orchestre .... (T)

(39) Jle regarde en me marrant et lui me sort une tête genre : "Bah quoi"? (T)

(40) A la fin, quand l'Américain sort de son char XD comment on a rit avec le bruit vraiment con ahah. Genre "pouh !" (T)

4 Distribution of construction types over the teenage and adult data

In this section we present the generalizations of this study by examining how the different constructions are distributed over the teenage and adult data.

First, we consider the quantitative results obtained within the pilot corpora PC-T and PC-A (see section 2 above). Table 1 gives the absolute numbers, relative frequencies and normalized frequencies of the constructions.

Total			Binominal			Post-determiner			Nominal Qualifier			Qualifying particle			Discourse marker			Quotative		
	n	N	n	%	N	n	%	N	n	%	N	n	%	N	n	%	N	n	%	N
T	71	59.1	2	3	1.5	31	43.5	25.8	4	5.5	3.3	28	39.5	23.3	2	3	1.6	4	5.5	3.3
A	27	22.5	0	0	0	22	81.5	18.3	2	7.5	1.6	2	7.5	1.6	0	0	0	1	3.5	0.8

Table 1: Absolute frequencies (n), relative frequencies (%) and normalized frequencies per 100,000 words (N) of lexical source construction and grammaticalized uses of *genre* in PC-T and PC-A

The most striking observation to emerge from the pilot corpora is that the word *genre* is only half as frequent in the similarly sized PC-A (51) as in PC-T (100). The grammaticalized uses are even less than half as common overall in PC-A than in PC-T. In 100,000 words of PC-T we find 57.5 grammaticalized uses of *genre* versus 22.5 in 100,000 words in PC-A. This is as such an interesting finding, which, despite the modest size of the pilot corpora, can be assumed to reflect a real general tendency, viz. that it is in the informal exchanges between teenagers, not between adults, that the highest frequency of grammaticalized uses of *genre* is found. As a reflection of the relative proportions of the distinct grammaticalized constructions, the pilot corpora can be expected to be somewhat less reliable. Still, if we compare the relative frequencies of the constructions for T and A within the pilot corpora (Table 1) and within the larger datasets (Table 2), we see a good degree of convergence, allowing us to conclude that in the adult data the postdeterminer construction forms the overwhelming majority (+/- 80%) which occurs with a normalized frequency around 18/100,000, while in the teenage data both the postdeterminer and qualifying particle uses are common (+/- 40%), occurring with a normalized frequency of around 25/100,000.

The more detailed comparison of the proportions of grammaticalized uses in teenage and adult data, then, we will base on the larger datasets. Table 2 gives both the absolute numbers and relative frequencies of the lexical source construction and grammaticalized uses within each dataset.

	Total		Binominal		Post-determiner		Nominal qualifier		Qualifying particle		Discourse particle		Quotative	
	n	%	n	%	n	%	n	%	n	%	n	%	n	%
T	482	100	9	2	179	37	16	3	211	44	17	3.5	50	10.5
A	484	100	6	1.25	375	77.5	39	8	54	11	1	0.25	9	2

Table 2: Distribution of lexical source construction and grammaticalized uses of *genre* across teenage (T) and adult (A) dataset

We first compare the relative frequencies of each construction type and then the proportions of NP-internal and NP-external grammaticalized uses.

The *binominal* construction occurred in comparable small proportions in the teenage (2%) and adult (1.2%) datasets, with nouns such as *musique*, *film*, *oeuvre d'art* and *institution* as N2. Clearly, taxonomizing subtypes is not the main discourse function of *genre* in these informal dialogic registers. The *postdeterminer* construction accounts for by far the largest portion (77.5%) of the adult uses of *genre* but for only 37.5 % of the teenagers' uses. This use, which creates generalizing cohesive relations, referring mostly to an exemplificatory antecedent, is arguably the most formal of all the grammaticalized uses of *genre*. In a study comparing the relative frequencies of the different uses of English *sort*, *kind*, *type* in the Times and in London teenage language, De Smedt, Brems & Davidse (2007) found that the postdeterminer construction predominated by far in the newspaper data. This formal character may explain why this use has caught on so strongly with adult speakers. *Nominal qualifying* constructions, in which *genre de* is a premodifier of N2, are less common in both the teenage (3%) and adult data (8%). But in the teenage data, this small fraction of 3% increases exponentially for the *qualifying particle* use to 44%. The reason for this discrepancy has, in our view, to be sought in the strong specialization in specific uses manifested by the main French type nouns with grammaticalized uses, *sorte*, *espèce* and *genre*. *Sorte* and *espèce* are not available as qualifying particles, but *genre* can take on this function and it does so with the high relative frequency of 44% in the teenage data investigated in this study. *Discourse particles* are relatively infrequent in our forum data, 3.5% in the teenage and 0.25% in the adult data, because their most typical locus is spontaneous speech. As markers of tentativity and hesitation, for instance, they require the forum writers to imitate casual speech. *Quotatives* account for a considerable portion (10.5%) of the teenage sample, but are marginal (2%) in the adult data. This is not surprising as innovative quotatives have been identified as a typical area of rapid change in the language of teenagers (Buchstaller 2006).

The distribution of the *NP-internal* and *NP-external* grammaticalized uses is particularly revealing. With the adults the NP-internal constructions predominate with 85.5%, while with the teenagers the NP-external uses have a majority of 58%. Clearly, strong innovation, detachment from NP-structure and creative semantic shift, is very much situated in the teenage data.

One important NP-external use is the quotative, which is five times more frequent in the teenage data (10.5%) than in the adult data (2%). As innovative quotatives are generally viewed as an indexical feature of teenage language (e.g. D'Arcy 2004, Tagliamonte & D'Arcy 2004), these figures seem easily explainable. *Genre* can be regarded as a marker of social identity of teenagers in the French-speaking world, comparable to English innovative quotatives such as *go*, *be like*, *be all*, and of course *(be) sort/kind of*. (But note that the latter are much less common (Vandelanotte & Davidse 2009) than French (*être/faire*) *genre*.)

The single biggest portion in the teenage data is formed by the qualifying particle use, 44%, which is four times more common than the fraction of 11% in the adult data. This high frequency is probably partly motivated by its functioning as a marker of teenage identity, like the quotative use. But, as noted above, teenage language is also leading the way here in making at least one type noun in French available as a qualifying particle, given the fact that *sorte* and *espèce* have not yielded qualifying particles.

The third NP-external use of *genre* is its discourse particle use, which, in its interaction with tentativity markers and hesitation phenomena, is very typically associated with spoken language. Our forum data, which remain informal written data with a strong interactional character, are less helpful and reliable here. We suspect that the small fractions found in our samples do not reflect the frequency of the discourse particle use in informal spoken language.

If we put together the comparison of the relative frequencies of each construction type and of the proportions of NP-internal and NP-external grammaticalized uses, we are struck by the *systematic* innovativeness of teenage usage. It leads the way in terms of progressive grammaticalization, and it does so in accordance with generally established tendencies, rooted in the semantic modules and structural layers of the language system. *Within* the NP, it has the postdeterminer use, which results from the binominal construction via *textual* subjectification, and the nominal qualifying use, which involves *expressive* subjectification in Traugott's (1989) terms. Within the uses that have *detached themselves from NP-structure*, the qualifying particle predominates numerically, followed by the quotative and then by the discourse particle use. In this way, this case study of *genre* strongly suggests that teenage language is highly relevant to the study of language change at large. In other words, systematic study of recent informal and spoken teenage language should be put higher on the research agenda of the diachronic-synchronic study of language change than it currently is (cf. also Caubet et al 2004).

This requires general availability of corpora of teenage language. Even though some – often smallish – corpora of teenage language exist, the great need for such data can at present only be met by Internet data such as the forum data used in this study, which had the added advantage of allowing easy comparison with very similar data from adult forums. The forum data are also relatively neat, yet well-contextualized and hence rather easy to manage in analysis. At the same time, they are not spoken language and therefore fail to represent uses strictly tied to spoken language. For the latter uses, data from chat sites would probably be very useful. In any case, it is our conviction that the study of language change has to set about studying teenage language more systematically and that, in order to do so, it will have to engage with Internet data, notwithstanding the methodological problems touched on in this study. We hope that this comparative study of the grammaticalized uses of *genre* in teenage and adult forum language plausibly underscores these general points.

References

Aijmer, Karen. 2002. *English Discourse Particles: Evidence from a Corpus*. Amsterdam: Benjamins.

Buchstaller, Isabelle. 2006. Diagnostics of age-graded linguistic behaviour: The case of the quotative system. *Journal of Sociolinguistics* 10: 3-30.

Caubet, Dominique, Jacqueline Billiez, Thierry Bulot, Isabelle Léglise and Catherine Miller. 2004. *Parlers jeunes, ici et là-bas: Pratiques et représentations.* Paris: L'Harmattan.

D'Arcy, Alex. 2004. Contextualizing St. John's Youth English within the Canadian quotative system. *Journal of English Linguistics* 32: 323-345.

Denison, David. 2002. History of the *sort of* construction family. Paper presented at the Second International Conference on Construction Grammar, University of Helsinki, 7 September 2002. [Online draft version available at http://lings.In.man.ac.uk/staff/dd/papers/sortof_iccg2.pdf.]

De Smedt, Liesbeth, Lieselotte Brems & Kristin Davidse. 2007. NP-internal functions and extended uses of the 'type' nouns *kind*, *sort*, and *type*: towards a comprehensive, corpus-based description. In Roberta Facchinetti (Ed.) *Corpus Linguistics 25 Years on*. Amsterdam: Rodopi, 225-255.

Diewald, Gabriele. 2006. Hecken und Heckenausdrücke – Versuch einer Neudefinition. In Emilia Calaresu, Cristina Guardiano und Klaus Hölker, eds. *Italienisch und Deutsch als Wissenschaftssprachen. Bestandsaufnahmen, Analysen, Perspektiven*. Berlin: LIT-Verlag (Romanistische Linguistik 7), 295-315.

Fleischman, Suzanne and Marina Yaguello. 1999. Discourse markers across languages ? Evidence from English and French. In Moder, Carol Lynn and Aida Martinovic-Zic, eds., *Discourse accross Languages and Cultures*. Amsterdam: Benjamins, 129–147.

Golato, Andrea. 2000. An innovative German quottive for reporting on embodied actions: *Und ich so/und er so* 'and I 'm like/and he's like'. *Journal of Pragmatics* 32 : 29-54.

Halliday, Michael. 1978. *Language as Social Semiotic: The Social Interpretation of Language and Meaning*, London: Edward Arnold.

Hopper, Paul J. and Elizabeth Closs Traugott. 2003. *Grammaticalization*. Cambridge: CUP.

Ito, Rika and Sali Tagliamonte. 2003. *Well* weird, *right* dodgy, *very* strange, *really* cool. *Language in Society* 32: 257-279.

Macaulay, Ronald. 2001. *You're like 'why not?'* The quotative expressions of Glasgow adolescents. *Journal of Sociolinguistics* 5: 3-21.

Macaulay, Ronald. 2006. Pure grammaticalization: The development of a teenage intensifier. *Language Variation and Change* 18: 267-283.

Martin, Jim. 1992. *English text: systems and structures*. Amsterdam: Benjamins.

McGregor, William. 1997. *Semiotic Grammar*. Oxford: Clarendon.

Tagliamonte, Sali & Alex D'Arcy. 2004. He's like, she's like: The quotative system in Canadian youth. *Journal of Sociolinguistics* 8: 493-514.

Traugott, Elizabeth Closs. 1989. On the rise of epistemic meanings in English: An example of subjectification in semantic change. *Language* 65: 31–55.

Traugott, Elizabeth Closs & Richard Dasher. 2002. *Regularity in semantic change*. Cambridge: CUP.

Vandelanotte, Lieven & Kristin Davidse. 2009. The emergence and structure of *be like* and related quotatives: a constructional account. *Cognitive Linguistics* 20: 777-807.

Van Peteghem, Marleen. 1995. Anaphores : marqueurs et interprétations. *Sémiotiques* 8: 57-78.

Willemse, Peter, Kristin Davidse, Lieselotte Brems. 2007. The development of extended type noun uses: a comparison between English *sort/kind* and French *sorte/espèce*. Paper presented at ICAME 28, Stratford-upon-Avon (23-27 May).

Data sources

Adojeunz, La communauté la plus cool du net, accessed on 16 and 17 December 2008, 12 and 13 July 2009.

http://www.adojeunz.com/forum/index.php

Discutons.org, *Forum de discussion*, accessed on 16 and 17 December 2008, 12 and 13 July 2009.

http://www.discutons.org/Debats_generaux_d_actualite-Forum-3.html

Is Part-of-Speech Tagging a Solved Task?
An Evaluation of POS Taggers for the German Web as Corpus

Eugenie Giesbrecht
FZI Research Center
for Information Technology
76131 Karlsruhe, Germany
giesbrecht@fzi.de

Stefan Evert
Institute of Cognitive Science
University of Osnabrck
49069 Osnabrück, Germany
stefan.evert@uos.de

Abstract

Part-of-speech (POS) tagging is an important preprocessing step in natural language processing. It is often considered to be a "solved task", with published tagging accuracies around 97%. Our evaluation of five state-of-the-art POS taggers on German Web texts shows that such high accuracies can only be achieved under artificial cross-validation conditions. In a real-life scenario, accuracy drops below 93% with enormous variation between different text genres, making the taggers unsuitable for fully automatic processing. We find that HMM taggers are more robust and much faster than advanced machine-learning approaches such as MaxEnt. Promising directions for future research are unsupervised learning of a tagger lexicon from large unannotated corpora, as well as developing adaptive tagging models.

1 Introduction

Automatic part-of-speech (POS) tagging is an important and widely-used preprocessing step in natural language processing applications, and it is almost indispensable for the exploitation of corpus data. At the same time, it is essentially considered a "solved task", with state-of-the-art taggers achieving per-word accuracies of 97%–98% (Schmid, 1995; Toutanova et al., 2003; Shen et al., 2007). While this still means that, on average, every other sentence contains a tagging error,[1] the accuracy is close to the level of agreement between human annotators and thus to the upper limit that can be expected from an automatic tagger.

Virtually all taggers have been trained and evaluated on newspaper text, though, and it is not clear whether they would achieve equally high accuracy on other genres such as spoken language, informal writing, or Web pages. The latter form a particularly important category in scientific research – where an increasing number of researchers turn to the World Wide Web as a convenient and inexhaustible source of natural language data (the "Web as Corpus" approach, see e.g. Kilgarriff and Grefenstette (2003)) – as well as commercial applications – where mining the Web for semantic knowledge, market intelligence, etc. has become one of the most successful applications of NLP technologies.

Therefore, the reported tagging accuracies of 97%–98% have to be understood as optimistic estimates, representing an ideal case for machine-learning approaches: (i) the taggers are applied to edited, highly standardized text with a low rate of errors and unusual patterns; and (ii) training and test data are very similar (usually from the same volume of the same newspaper), so that overfitting of the training data is rewarded to a certain degree.

The goal of this paper is to find out whether the published tagging accuracies – which are often taken for granted by researchers and developers using off-the-shelf POS taggers in their NLP systems – can also be achieved under real-life conditions, where taggers have to deal with less standardized genres such as Web texts. Our hypothesis is that the quality of POS tagging will be dramatically reduced under such circumstances, perhaps even to a degree that makes its usefulness as a general preprocessing step questionable.[2] In order to test this hypothesis, we evaluate five state-of-

[1] With a per-word tagging accuracy of 97%, there is a probability of 45.6% that a 20-word sentence (the average sentence length in the Brown corpus) contains one or more tagging errors.

[2] With a per-word accuracy of 92%, less than one in five sentences will be error-free. Some sources also claim that the baseline accuracy achieved by a simple most-frequent-tag heuristic can be as high as 90% under favourable conditions, cf. (Manning and Schütze, 1999, 372).

the-art statistical taggers on a representative collection of German Web texts sampled from the DEWAC corpus (Baroni et al., 2009). Since we are not aware of any systematic comparative evaluation of German POS taggers, we also determine "ideal" tagging accuracies by cross-validation on the TIGER treebank (Brants et al., 2002), to be used as a point of reference.

The rest of the paper is organized as follows. Section 2 gives an overview of the state of the art in statistical POS tagging and lists published evaluation results for German. Section 3 describes our evaluation methodology and the corpora used in our experiments. Evaluation results are given in Section 4, with a qualitative analysis of tagging errors in Section 5. Section 6 examines how tagging accuracy is influenced by tagset granularity and the genre of a Web page. The main insights we have obtained for the development of more robust POS taggers are summarized in Section 7.

2 State-of-the-art taggers for German

Most POS taggers have been developed for English, using the Penn Treebank (Marcus et al., 1993) as training and evaluation data. The best published tagging accuracies fall into a narrow range from 96.50% to 97.33% (Brants, 2000; Toutanova et al., 2003; Giménez and Màrquez, 2004; Shen et al., 2007). While the rule-based EngCG tagger is reported to achieve very high accuracy in combination with a statistical disambiguator (Tapanainen and Voutilainen, 1994), it is only available as a commercial product and has therefore been excluded from our study.

However, these high accuracy figures have to be qualified for two reasons. First, there are some doubts about the consistency of the Penn Treebank annotation (Dickinson and Meurers, 2003). Second, the proportion of unknown words is very low in all reported evaluation experiments (ca. 2%). It is not clear whether comparable results would be achieved for a text genre with richer, less controlled vocabulary (such as Web pages) or a language with more complex and productive morphology (such as German).

There are only few published evaluation results for German POS taggers, summarized in Table 1. The top two rows show accuracies reported by the developers of the two most widely-used statistical taggers for German, TnT (Brants, 2000) and TreeTagger (Schmid, 1995). Both are in the same range as state-of-the-art English taggers, and TreeTagger even outperforms the best current tagger for English. These results are not directly comparable, though, since they have been obtained on different gold standards – TnT was trained and evaluated on the NEGRA treebank (Skut et al., 1998), TreeTagger on a proprietary gold standard.

	overall	UW	KW	% unk.
TnT	96.70	89.0	97.7	11.9
TreeTagger	97.53	78.0	97.4	2.0
TBL	94.57	81.5	—	15.0
TreeTagger	95.27	84.1	—	15.0

Table 1: Published evaluation results of German POS taggers (UW = accuracy on unknown words, KW = accuracy on known words, % unk. = proportion of unknown words; all values are percentages). The top rows show results reported by the original developers (Brants, 2000; Schmid, 1995), the bottom rows show results from a comparative evaluation study (Volk and Schneider, 1998).

As expected, the proportion of unknown words (12%–15%) is much higher than for the English taggers. Note that TreeTagger makes use of a heuristic lexicon extracted from a large, automatically tagged corpus (Schmid, 1995, Sec. 3.3). This lexicon reduces the proportion of unknown words to only 2%, similar to the Penn Treebank, and is also included in the standard parameter file distributed with TreeTagger (cf. Sec. 3). When Volk and Schneider (1998) re-train the tagger without such a heuristic lexicon, the proportion of unknown words increases to 15%.

The bottom rows of Table 1 show results from an independent evaluation study (Volk and Schneider, 1998), comparing TreeTagger with Brill's (1995) transformation-based learning approach (TBL). The accuracy of TreeTagger is much lower than reported by Schmid (1995) – only 95.27% *vs* 97.53% – and falls behind the English state of the art. While differences in the training regime may account for part of the decrease, the most important factor is certainly the higher proportion of unknown words (15% *vs* 2%) resulting from the lack of a heuristic lexicon. Still, the statistical approach of TreeTagger outperforms the rule-based TBL tagger and is also computationally more efficient with a training time of less than 2 minutes *vs* approx. 30 hours for TBL (Volk and Schneider, 1998, Sec. 2).

2.1 Taggers selected for the evaluation

We decided to restrict our evaluation to statistical taggers, which achieve the best published results for both English and German. Likewise, only freely available implementations – which could easily be trained and evaluated on our data, and are most widely used by researchers and developers – were taken into consideration. In addition to the best-performing German taggers (TnT and TreeTagger), we included three further state-of-the-art taggers, resulting in the following list of candidates:

1. TreeTagger[3] – HMM tagger using decision trees for smoothing; best published tagging accuracy for German; widely used by researchers due to its easy availability (Schmid, 1995);
2. TnT – another widely-used HMM tagger, with standard smoothing (Brants, 2000);
3. SVMTagger – open-source tagger using support vector machines for classification (Giménez and Màrquez, 2004);
4. Stanford tagger – bidirectional MaxEnt tagger with the best published tagging accuracy for English (Toutanova et al., 2003);
5. Apache UIMA Tagger[4] – open-source HMM tagger written in Java, implemented by one of the authors (see below for details).

2.2 The UIMA Tagger

The UIMA Tagger closely follows the standard HMM approach described by Brants (2000), omitting some advanced heuristics that are used by the TnT implementation but not mentioned in the paper. Like TnT, the UIMA Tagger is based on a trigram Hidden Markov Model, with trigram probabilities smoothed by deleted interpolation. Lexical probabilities of unknown words are guessed from suffix strings, estimated from words that occur less than 10 times in the training corpus. Separate suffix probabilities are computed for captialized and non-capitalized words, since capitalization provides an important morphological cue in German (all common nouns are captialized).

For known words, only the tags available in the model are used for prediction; otherwise Ukkonen suffix trees (Ukkonen, 1995) are used to find the longest suffix of an unknown word for which a suffix probability has been estimated. No further heuristics and smoothing strategies are implemented in the current version of the UIMA Tagger.

The UIMA Tagger was included in our evaluation because it provides an excellent open-source platform for experiments on improving tagging accuracy, while other HMM taggers such as TnT and TreeTagger are only available in the form of binary packages. The recent open-source implementation HunPos (Halácsy et al., 2007) is written in OCaml, which has a much smaller user base than Java. Last but not least, the UIMA Tagger is natively supported in Apache UIMA[5] (Unstructured Information Management Architecture), a framework for industrial text analytics applications that is also being used by an increasing number of NLP and Information Retrieval researchers (Müller et al., 2008; Nyberg et al., 2008). Together with its permissive Apache License, this will encourage academic and industrial research groups to adapt the tagger to their special requirements (such as processing Web pages), and to contribute their improvements back to the open-source code base.

3 Evaluation methodology

Since no directly comparable evaluation results have been published for German POS taggers, we first evaluated all five taggers listed in Section 2.1 on the TIGER treebank (Brants et al., 2002), which is currently the largest manually annotated German corpus. It consists of about 900,000 tokens (50,000 sentences) of German newspaper text, taken from the *Frankfurter Rundschau*.[6] Each sentence has been annotated with manually validated POS tags, lemmas, morphosyntactic features and parse trees. Annotations were carried out by two independent annotators, followed by a consistency check (Brants and Hansen, 2002). For our purposes, only the POS annotation according to the STTS tagset (Schiller et al., 1999) was used.

The evaluation was carried out by 10-fold cross-validation. We divided the corpus into 10 contiguous parts, which we consider to be a slightly more realistic setting than taking every tenth sentence

[3] Binary packages for Linux, Solaris, Mac OS X and Windows can be downloaded from `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html`, together with pre-compiled parameter files for 8 different languages.

[4] The UIMA Tagger can be downloaded from `http://incubator.apache.org/uima/sandbox.html#tagger.annotator`

[5] `http://incubator.apache.org/uima/`

[6] The token counts given in this paper include all tokens, i.e. words, numbers and punctuation.

or choosing random sentences. Then, each tagger was trained on 9 of the 10 parts (using standard settings for all meta-parameters) and evaluated on the held-out part. In Section 4.1, we report the mean and standard deviation of per-word tagging accuracy across all 10 cross-validation folds.[7] This evaluation setup is very similar to published evaluation experiments for TnT, TreeTagger, and the English POS taggers. It provides a fair comparison of the five different taggers and serves as a point of reference for our main evaluation experiment on Web texts. One has to keep in mind, though, that – like in most other published evaluation studies – the POS taggers are evaluated on text that is very similar to their training data, which will rarely be the case in real-world applications.

Finally, all taggers are trained on the complete TIGER treebank. The resulting parameter files are used for all further evaluation experiments, ensuring a fair comparison between the taggers. In addition, we evaluate the standard parameter files (SPF) distributed with TnT and TreeTagger, which many researchers use for convenience.

Since no manually annotated Web reference corpus is available, we had to compile our own gold standard for the evaluation on Web text. For this purpose, we selected a random sample of Web pages from DEWAC (Baroni et al., 2009), a German Web corpus containing approx. 1.6 billion tokens of text collected in the year 2005.[8] The DEWAC corpus was cleaned by removing duplicate pages and so-called boilerplate (automatically generated page content such as navigation bars, advertising and legal disclaimers). It was then tagged and lemmatized using TreeTagger with its standard parameter file for German. See Baroni et al. (2009) for details on the corpus preparation.

Our gold standard consists of 13 Web pages from widely different genres, amounting to a total of 10,057 tokens of text. We manually corrected the automatic tokenization and POS tagging provided by the DEWAC corpus, using the same STTS tagset as in the TIGER treebank. In Section 4.2, we report the per-word accuracy achieved by each of the five taggers on the DEWAC gold standard, using the TIGER treebank for training (as well as SPF for TnT and TreeTagger). Since these results are not obtained through a cross-validation scheme, it is not meaningful to calculate standard deviation (but see Section 6.1 for the variability of tagging accuracy across text genres).

4 Evaluation results

4.1 TIGER treebank

The top row of Table 3 shows the mean and standard deviation of per-word tagging accuracy on the TIGER treebank for all selected taggers, obtained by 10-fold cross-validation as described in Section 3. The other rows give separate accuracy figures for known and unknown words, as well as the percentage of unknown words in the test data. Accuracies obtained with the standard parameter files of TnT and TreeTagger are shown in Table 2.[9]

Tagging with the standard parameter file of TreeTagger results in a per-word accuracy of 95.82%, which is 1.71% less than the value reported by Schmid (1995). The accuracy of TnT is also considerably lower than the published figure. In the cross-validation experiment (Table 3), where training and test data are from the same corpus, both taggers achieve considerably better accuracy, though TreeTagger still falls short of the published value of 97.53% (probably due to the lack of a heuristic lexicon in our experiments).

	overall	KW	UW	% unk.
TreeTagger	95.82	96.27	79.88	2.7
TnT	95.71	96.97	86.94	12.6

Table 2: Per-word tagging accuracy on the TIGER treebank, using the standard parameter files (SPF) distributed with TreeTagger and TnT.

The best result in the cross-validation experiment was achieved by the bidirectional MaxEnt Stanford tagger, whose mean total accuracy of 97.63% matches the published figure for TreeTagger, making it the best known POS tagger for German text. It is also remarkable that this total accuracy is as high as the known-words accuracy of TreeTagger and TnT. Second place is achieved by the SVM tagger. The Stanford tagger is significantly better than all HMM taggers (paired t-test against TnT/TreeTagger: t=11.33, df=9, p<.001) and the SVM tagger (paired t-test: t=13.21, df=9,

[7] Since all folds contain approximately the same number of tokens, this macro-averaged mean is equal to the micro-averaged per-word accuracy on the full corpus.

[8] Note that Baroni et al. (2009) report a much smaller size of approx. 1.28 billion tokens, because their counts exclude punctuation, numbers and other non-word tokens (Baroni et al., 2009, Sec. 3.4).

[9] Since these results have not been obtained by cross-validation, standard deviation is not available.

	TreeTagger	Stanford	UIMA	TnT	SVM
total accuracy (%)	96.89±0.34	**97.63±0.24**	96.04±0.38	96.92±0.31	97.12±0.20
known words (%)	97.62±0.21	–	97.50±0.18	97.59±0.20	97.71±0.17
unknown words (%)	87.89±0.99	**91.66±0.83**	79.59±1.30	89.16±0.85	90.16±0.84
% of unknown words	7.44±0.78	7.52±0.46	8.10±0.71	7.85±0.88	7.82±0.82

Table 3: Evaluation results for 10-fold cross-validation on the TIGER treebank. For each tagger, we report mean and standard deviation of per-word accuracy across the 10 folds (all values are percentages).

p<.001). This is mostly due to its significantly higher accuracy on unknown words.

The high accuracy of the Stanford tagger comes at a price, though, due to the computational complexity of its advanced statistical model. Tagging the 900,000 tokens of the TIGER treebank takes more than 45 minutes with the Stanford tagger, compared to less than 10 seconds with TreeTagger (measured on 2.6 GHz Dual Core AMD Opteron 285 Processor). Likewise, training the Stanford tagger on TIGER took approx. 5.5 hours, while the TreeTagger completed its supervised training procedure in less than 10 seconds. The gain in accuracy of approx. 0.7% compared to the best HMM tagger is relatively small, and it is presumably worth its while in case achieving the best possible accuracy is crucial for the task at hand.

4.2 Web texts

For this experiment, we trained all taggers on the complete TIGER treebank and then evaluated their performance on DEWAC, in order to simulate a realistic setting where no in-domain training data are available and a standard parameter file trained on a newspaper corpus has to be used. Evaluation results are shown in Table 4; the first column lists the results obtained by TreeTagger with its standard parameter file (labelled TT-SPF).

Disregarding the TT-SPF data, we see that the best overall accuracy is now achieved by TnT, a HMM-based tagger. While the Stanford tagger is considerably better than its competitors on unknown words, its overall accuracy falls slightly short of TnT.[10] These results clearly indicate a certain degree of overtraining for the machine-learning approaches (Stanford and SVM tagger), while TnT generalizes better to less standardized genres such as Web texts. We may thus conclude that HMM-based approaches are both more robust and computationally more efficient than MaxEnt and other advanced machine-learning techniques.

Surprisingly, TreeTagger performs worse than all other taggers if it is trained on the TIGER treebank; the reasons for this discrepancy are not entirely clear yet. When used with its standard parameter file (SPF), on the other hand, it achieves a much higher accuracy than TnT (93.71% *vs* 92.69%). This appears to be due to the inclusion of a heuristic tagger lexicon in the SPF, which reduces the proportion of unknown words to 4.15%, compared to 13.44% for TnT.

On the whole, there is a dramatic decrease in accuracy for all taggers under real-life conditions, caused (amongst others) by a much higher proportion of unknown words than in the cross-validation experiment. The unknown words in Web texts also seem to be more “difficult” than those in TIGER, so that e.g. the unknown-words accuracy of the Stanford tagger drops from 91.66% to 75.35%. The most robust results are achieved by TreeTagger with its standard parameter file, but a per-word accuracy of 93.71% is still unsatisfactory for most applications in linguistics and NLP.

5 Qualitative error analysis

A closer look at the error statistics for individual tags – using the best-performing tagger on DEWAC, i.e. TreeTagger with its SPF, as an example – revealed similar error sources as reported by Schmid (1995) and Volk and Schneider (1998). Most of the errors can be traced to insufficient distributional differences within major categories (e.g., proper *vs* common nouns or finite *vs* infini-

[10] It is difficult to determine whether the observed differences are significant, since these data have not been obtained from a cross-validation procedure. In view of the enormous variation between individual texts in the DEWAC gold standard (see discussion in Section 6.1), it is clearly inappropriate to pool all data into a sample of 10,057 tokens. Paired t-tests across the 13 individual texts find significant differences (wrt. macro-averaged accuracy as shown in Table 7) only between TnT and TreeTagger (as well as TT-SPF and TnT), again due to the large variation between texts.

	TT-SPF$^{a)}$	TT$^{b)}$	Stanford	UIMA	TnT	SVM
total accuracy (%)	**93.71**	90.78	92.61	91.68	**92.69**	92.36
known words (%)	95.42	93.59	—	95.59	95.90	95.91
unknown words (%)	54.30	69.12	**75.35**	66.49	71.99	69.45
% of unknown words	4.15	11.48	13.00	13.43	13.44	13.43

[a] TreeTagger with standard parameter file included in distribution
[b] TreeTagger with parameter file trained on the TIGER treebank

Table 4: Evaluation results on the DEWAC gold standard. All taggers have been trained on the complete TIGER treebank for this experiment (except for TT-SPF).

tive verbs) or between certain categories (e.g., adverbs *vs* adverbially used adjectives).

TIGER	DEWAC	TIGER	DEWAC
NE	**$(**	ADJD	AVD
APPR	NE	ADJA	**XY**
VVFIN	**FM**	PIS	**CARD**
ADV	NN	VVINF	ADJA
NN	VVFIN	VVPP	APPR

Table 5: Most frequently misclassified POS tags in TIGER and DEWAC (TreeTagger with SPF).

Table 5 shows the gold standard POS tags that were misclassified most frequently. Apart from typical tagging errors for the main parts of speech such as nouns and verbs, there are a number of unexpected tags among the 10 most frequent error types on Web texts: $((sentence-internal punctuation, except for comma), FM (foreign material), CARD (cardinal numbers) and XY (special characters). All of these are prevalent in Web texts, and they appear to be an important factor behind the low tagging accuracy.

The comparison of the most frequent tag confusion pairs for TIGER and DEWAC (see Table 6) confirms our intuition that – in addition to well-known problems (Schmid, 1995; Volk and Schneider, 1998) that were confirmed by our TIGER experiments – there are many "new" error types due to the confusion of punctuation signs, foreign words and cardinals with common nouns, proper nouns and adjectives.

TIGER Treebank		DEWAC	
correct tag	TT-SPF	correct tag	TT-SPF
NE	NN	NE	NN
APPR	KOKOM	**$(**	**$.**
NN	NE	**FM**	**NN**
VVINF	VVFIN	NN	NE
VVFIN	VVPP	**FM**	**NE**
ADJA	NN	**CARD**	**NN**
PWAV	KOUS	**$(**	**ADJA**
ADV	ADJD	ADV	ADJD
ADJD	ADV	**XY**	**NE**
VVFIN	VVINF	VVFIN	VVPP

Table 6: Most frequently confused POS tags.

6 Determinants of tagging accuracy

6.1 Text genre

The Web pages included in our DEWAC gold standard represent entirely different text genres. This allowed us to test whether the low overall tagging accuracy in Table 4 reflects a general difficulty of processing Web data, or whether there are "easier" and "harder" genres on the Web. Table 7 shows separate per-word accuracy results for each genre.

In 7 out of 13 genres, TreeTagger with its standard parameter file (TT-SPF) achieves state-of-the-art accuracy between 95.42% and 98.25%. These "easy" genres include various news reports, a political speech, a support programme announcement, and other types of expository prose – all quite similar to typical newspaper text. In most cases, the percentage of unknown words is also very low (details omitted for space reasons).

Clearly, there are four problematic genres, where the accuracy of all taggers falls below 94%: an episode guide for a TV series, postings from an online forum, a conference information site,[11] and a news report on the archbishop of Boston (highlighted in italics in Table 7). Except for the latter, these are Web-specific text genres that have not been carefully edited like the newspaper articles in the TIGER treebank. As a result, they contain many typographical and grammatical mistakes, as well as tabular listings. The highest concentra-

[11] Reassuringly, this is not a computational linguistics conference, but rather an annual meeting organized by a psychotherapy journal.

	Genre	TT-SPF[a)]	TT[b)]	TnT	Stanford	SVM	UIMA
1.	*TV episode guide*	***93.89***	*90.87*	*92.79*	*92.83*	*92.78*	*89.91*
2.	news report (medicine)	**96.88**	97.12	95.92	96.16	95.68	94.26
3.	political speech	**97.52**	96.56	96.42	96.15	93.81	95.61
4.	job market news	**97.46**	93.65	96.19	96.95	95.18	95.44
5.	story (Paul of Thebes)	**95.42**	94.84	95.08	95.37	95.08	93.87
6.	exposition programme	**94.23**	92.13	92.83	92.66	93.01	90.75
7.	*online forum*	***88.01***	*79.97*	*85.56*	*84.47*	*84.51*	*84.47*
8.	report on infections	**98.25**	96.89	97.28	**98.25**	97.08	95.54
9.	*conference information*	*90.98*	*89.18*	*92.01*	*90.98*	***93.30***	*92.55*
10.	IT news (CeBIT)	93.69	92.73	92.93	94.07	94.07	**95.42**
11.	info (support programme)	97.10	98.51	98.01	**99.50**	97.01	98.02
12.	*news report (archbishop)*	*91.97*	*87.15*	*91.97*	*91.97*	***93.97***	*90.80*
13.	synopsis of cold war	96.67	94.86	96.49	95.68	95.40	**97.30**
		94.77 ±3.04	92.65 ±5.04	94.11 ±3.31	94.23 ±3.85	93.91 ±3.15	93.38 ±3.67

[a] TreeTagger with standard parameter file included in distribution
[b] TreeTagger with parameter file trained on the TIGER treebank

Table 7: Tagging accuracies for the different text genres in the DEWAC gold standard. Note that the macro-averaged means in the bottom row are different from the micro-averaged means shown in Table 4. The best result for each genre is highlighted in bold font; particularly difficult genres are printed in italics.

tion of tagging errors was found in a forum posting written entirely in lowercase by a non-native speaker, as the following excerpt shows:[12]

> ... *hallo*$_{\mathrm{ITJ}}$ *meine*$_{\mathrm{PPOSAT}}$ ***name***$_{\mathbf{NN}}$ *ist*$_{\mathrm{VAFIN}}$ ***nesko***$_{\mathbf{ADJD}}$,$_{\mathrm{\$,}}$ *wohne*$_{\mathrm{VVFIN}}$ *in*$_{\mathrm{APPR}}$ ***dubrovnik***$_{\mathbf{NN}}$ *in*$_{\mathrm{APPR}}$ ***kroatien***$_{\mathbf{NN}}$.$_{\mathrm{\$.}}$ *habe*$_{\mathrm{VAFIN}}$ *schon*$_{\mathrm{ADV}}$ ***stones***$_{\mathbf{ADJA}}$ ***karte***$_{\mathbf{NN}}$ *fur*$_{\mathrm{XY}}$ ***olympia***$_{\mathbf{ADJD}}$ ***stadion***$_{\mathbf{ADJA}}$ ***konzert***$_{\mathbf{NN}}$ *und*$_{\mathrm{KON}}$ *mochte*$_{\mathrm{VVFIN}}$ *gerne*$_{\mathrm{ADV}}$ *auch*$_{\mathrm{ADV}}$ *fur*$_{\mathrm{XY}}$ ***halle***$_{\mathbf{VVFIN}}$ ...

The author of this text fails to capitalize names and common nouns (highlighted in bold font) and omits the diaresis in words like *für* and *möchte* (underlined). As a result, almost every other word is not recognised by the tagger, resulting in an accuracy of only 58% for this sentence. There are also various grammatical mistakes, which would pose additional difficulties for the taggers even if there were no unknown words.

Table 7 shows that there is no single best tagger for Web texts that works equally well across all genres. Different heuristics and optimizations used by individual taggers make them particularly suitable for specific text genres. TreeTagger with its standard parameter file achieves the best accuracy for 8 out of 13 genres and works reasonably well for the remaining 5 genres. It is therefore the recommended choice for Web texts and other non-standardized genres at the current time.

6.2 Tagset granularity

Applications of Web corpora may not always require the full detail of the 54 different tags in the STTS tagset (examples include basic information mining, computational lexicography, and distributional semantic models). In such cases, a coarse-grained tagset that distinguishes, e.g., verbs from nouns and adjectives, will be sufficient. In this section, we show that mapping parts of speech to such a reduced tagset results in substantially higher tagging accuracy. Again, we use the best-performing tagger on Web texts, TreeTagger with its standard parameter file, as an example.

The TIGER treebank and the DEWAC gold standard were first tagged with the original STTS tagset (54 tags), then we mapped the output of the tagger onto a reduced tagset (14 tags for major parts of speech) before carrying out the evaluation. Tagging accuracy increases by almost 2% on TIGER, and almost 3% on the Web texts (see Table 8). There is also a drastic increase in unknown-words accuracy (by ca. 8%–14%), as many confusion pairs are now mapped to the same coarse POS tag. In particular, the most frequent errors type specific to Web texts disappear completely or are considerably reduced.

Table 9 shows separate accuracy results for each text genre in the DEWAC gold standard, using the

[12] The POS tags in this excerpt were automatically assigned by TreeTagger with its standard parameter file.

	overall	KW	UW	% unk.
TIGER treebank (TT, SPF)				
fine	95.82	96.27	79.88	2.70
coarse	97.79	97.80	93.50	2.70
TIGER treebank (TT, cross-validation)				
fine	96.90	97.62	87.89	7.40
coarse	98.28	98.50	95.60	7.40
DEWAC *gold standard (TT, SPF)*				
fine	93.71	95.42	54.30	4.15
coarse	96.51	97.81	66.50	4.15

Table 8: TreeTagger accuracy on TIGER and DEWAC for fine *vs* coarse tagset.

reduced tagset as described above. The gain in accuracy ranges from ca. 1% (for "easy" genres) up to almost 6% for particularly difficult texts. Even the online forum postings can now be tagged with an accuracy of 93.75%.

	fine tagset		coarse tagset	
#	all	unknown	all	unknown
1	*93.89*	*52.63*	*96.16*	*64.47*
2	96.88	85.71	99.04	92.85
3	97.52	58.33	98.21	58.33
4	97.46	80.00	97.97	80.00
5	95.42	68.62	97.28	72.55
6	94.23	73.91	97.90	95.65
7	*88.01*	*39.20*	*93.75*	*57.60*
8	98.25	100.00	99.42	100.00
9	*90.98*	*33.33*	*94.33*	*43.33*
10	93.69	46.42	95.79	57.14
11	97.10	33.33	99.50	100.00
12	*91.97*	*92.85*	*97.19*	*92.85*
13	96.67	27.27	97.02	36.26
	94.77	60.89	97.20	73.16
	±3.04	±24.44	±1.80	±22.01

Table 9: Comparison of tagging accuracy for fine and coarse tagset across DEWAC text genres (TT-SPF). "Difficult" genres are displayed in italics.

7 Conclusions

The goal of the study reported here was to empirically evaluate the performance of POS taggers trained on newspaper corpora in a real-world scenario, esp. when applied to less standardized text genres such as Web pages. Since there is no suitable Web reference corpus, we annotated a sample of German Web pages from the DEWAC corpus using a semi-automatic procedure. Five state-of-the art statistical taggers were trained on the TIGER treebank and evaluated on the new DEWAC gold standard.

Cross-validation on TIGER established the MaxEnt-based Stanford tagger as the best-performing tagger for German under the artificial "ideal" conditions used by most evaluation studies. Its per-word accuracy of 97.63% exceeds the published TreeTagger result of 97.54%, at the cost of much higher computational complexity (by more than a factor of 300).

When applied to Web texts, the accuracy of all taggers drops drastically, e.g. from 97.63% to 92.61% for the Stanford tagger. It is also no longer the best tagger in this scenario, being outperformed by the best HMM-based tagger TnT (92.69%). We take this result as an indication of overfitting by advanced machine-learning techniques such as MaxEnt and SVM. Surprisingly, TreeTagger achieves the lowest accuracy of all five taggers in the comparative DEWAC evaluation. Using the standard parameter file included in its distribution (which contains a heuristic lexicon extracted from a large, automatically tagged corpus), TreeTagger outperforms TnT by a margin of 1%. Its per-word accuracy of 93.71% is still not adequate for most applications, though.

A closer look at the individual texts of the DEWAC gold standard revealed that certain "easy" genres of Web pages can be tagged with state-of-the-art accuracy. Other, Web-specific genres such as online forum postings are "hard" and may result in tagging accuracies below 90%. If only a coarse-grained distinction between major parts of speech is required, a tagging accuracy of up to 96.51% can be achieved. Such a mapping to a reduced tagset is particularly beneficial for the "hard" Web genres, which can then be tagged with satisfactory accuracy (93.75% *vs* 88.01%).

We realize that making the task easier by reducing the number of tags is not an ultimate goal. The adaptation of statistical models for cross-domain tagging is currently a *hot topic* in NLP research (Finkel and Manning, 2009; Daumé III, 2009). Based on the insights from the latter and our in-depth study of POS taggers, we plan to develop more robust taggers for the Web.

Acknowledgments

This work was partially supported by German "Federal Ministry of Economics" (BMWi) under the project Theseus (number 01MQ07019).

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky Wide Web: A collection of very large linguistically processed Web-crawled corpora. *Language Resources and Evaluation*. To appear.

Sabine Brants and Silvia Hansen. 2002. Developments in the tiger annotation scheme and their realization in the corpus. In *Proceedings of the Third Conference on Language Resources and Evaluation (LREC 2002)*, pages 1643–1649.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT 2002)*, Sozopol, Bulgaria.

Thorsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing*, pages 224–231.

Eric Brill. 1995. Unsupervised learning of disambiguation rules for part of speech tagging. In *Proceedings of the Third ACL Workshop on Very Large Corpora*, pages 1–13.

Hal Daumé III. 2009. Bayesian multitask learning with latent hierarchies. In *Conference on Uncertainty in Artificial Intelligence*, Montreal, Canada.

Markus Dickinson and Walt Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*, pages 107–114, Budapest, Hungary.

Jenny Rose Finkel and Christopher D. Manning. 2009. Hierarchical bayesian domain adaptation. In *Proceedings of the North American Association of Computational Linguistics (NAACL 2009)*.

Jesús Giménez and Lluís Màrquez. 2004. SVMTool: A general POS tagger generator based on support vector machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal.

Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. HunPos – an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Posters and Demonstrations Sessions*, pages 209–212, Prague, Czech Republic.

Adam Kilgarriff and Gregory Grefenstette. 2003. Introduction to the special issue on the Web as corpus. *Computational Linguistics*, 29(3):333–347.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330.

Christof Müller, Torsten Zesch, Mark-Christoph Müller, Delphine Bernhard, Kateryna Ignatova, Iryna Gurevych, and Max Mühlhuser. 2008. Flexible UIMA components for information retrieval research. In *Proceedings of the LREC 2008 Workshop 'Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP'*, pages 24–27, May.

E. Nyberg, E. Riebling, R.C. Wang, and R: Frederking. 2008. Integrating a natural language message preprocessor with uima. In *Proceedings of the LREC 2008 Workshop 'Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP'*, pages 28–31, May.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, IMS, University of Stuttgart and SfS, University of Tübingen, August.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to German. In *Proceedings of the ACL SIGDAT Workshop*, March.

Libin Shen, Giorgio Satta, and Aravind Joshi. 2007. Guided learning for bidirectional sequence classification. In *ACL*. The Association for Computer Linguistics.

Wojciech Skut, Thorsten Brants, Brigitte Krenn, and Hans Uszkoreit. 1998. A linguistically interpreted corpus of German newspaper texts. In *Proceedings of the ESSLLI Workshop on Recent Advances in Corpus Annotation*, Saarbrücken, Germany.

Pasi Tapanainen and Atro Voutilainen. 1994. Tagging accurately – Don't guess if you know. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, pages 47–52.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

E. Ukkonen. 1995. On-line construction of suffix trees. *Algorithmica*, 14(3):249–260.

Martin Volk and Gerold Schneider. 1998. Comparing a statistical and a rule-based tagger for German. In *Proceedings of the 4th Conference on Natural Language Processing*, KONVENS-98, pages 125–137, Bonn.

Looking for French deverbal nouns in an evolving Web (a short history of WAC)

Nabil Hathout **Franck Sajous** **Ludovic Tanguy**
CLLE / CNRS and University of Toulouse, France
{hathout,sajous,tanguy}@univ-tlse2.fr

Abstract

This papers describes an 8-year-long research effort for automatically collecting new French deverbal nouns on the Web. The goal has remained the same: building an extensive and cumulative list of noun-verb pairs where the noun denotes the action expressed by the verb (e.g. *production* - *produce*). This list is used for both linguistic research and for NLP applications. The initial method consisted in taking advantage of the former Altavista search engine, allowing for a direct access to unknown word forms. The second technique led us to develop a specific crawler, which raised a number of technical difficulties. In the third experiment, we use a collection of web pages made available to us by a commercial search engine. Through all these stages, the general method has remained the same, and the results are similar and cumulative, although the technical environment has greatly evolved.

1 Introduction

The Web has been successfully used as a corpus for more than 10 years now, and as everything web-related, things have been evolving at tremendous speed. From the pioneer hackings of early search-engines in the late 20th century to the current development of linguistically-aware web corpus builders, many different efforts have been made to tap into this bottomless pit of linguistic data. What we present here is the technical evolutions of a narrow-focused research effort we have been working on for about 8 years. Our goal is the automatic acquisition of new French words, to be used as descriptive materials for morphology, and to a certain extent as a resource for natural language processing. More precisely, we search for new suffixed word forms, based on a set of productive French suffixes: mainly -*age*, -*ion* and -*ment*, which are used to coin nouns from verbs. Section 2 describes more precisely our objectives.

Although this task is quite simple with regards to current techniques in traditional corpus linguistics, complications arise when it is applied to the Web, as noted by Lüdeling et al. (2007). The main problem is that we are looking for word forms we know to be quite rare, and for which we only know the ending substring. If the Web is a very good answer to the former characteristic (because of its size and constant evolution), it is not adapted to the latter. This led us to use three different techniques for getting to our end. Each change from one technique to the other can be explained by the evolution of Web access. Section 3 describes the main method we used. In section 4, we try to draw a short history of the main evolutionary steps in using the Web as a corpus. Finally, section 5 describes more technically the three different solutions we applied along the last 8 years and the corresponding results.

2 The quest for French derived words

2.1 Data for NLP and extensive morphology

There is a large number of inflexional lexica available for many languages but very few derivational ones. For instance, we only know of two morphological databases for English: CELEX (Baayen et al., 1995) and Catvar (Habash and Dorr, 2003). CELEX also includes databases for German and Dutch. For French, hardly any such database exists. One exception is Verbaction[1] which describes the deverbal nouns of a large set of French verbs.

Derivational databases have initially been set up and used by psycho-linguists working on the mental lexicon and on the processing of derived words. They have also been used in NLP applications and Information Retrieval experiments. For instance, the French parser Syntex (Bourigault and Fabre, 2000) uses Verbaction for the disambiguation of PP attachments and Jing and Tzoukerman (1999) propose a method of query expansion with morphologically related words from CELEX. Derivational resources are also used in linguistics as corpora for the description of morphological pro-

[1] w3.erss.univ-tlse2.fr/verbaction/

cesses. These resources must be very large in order to allow for the observation and study of rare phenomena. This approach is known as "extensive morphology." Morpho-phonological studies such as (Plénat, 2000) or morpho-semantic ones such as (Hathout et al., 2003) have shown the fruitfulness of this approach and how the use of great quantity of data leads to new insights on the morphological phenomena (see (Hathout et al., 2008) for a detailed presentation of extensive morphology).

In order to study a given morphological phenomenon, say the effect of the length of a stem on the truncation of its final rhyme (for instance, why is the stem truncated in *inoxydation* 'process that makes steel become stainless' which should be *inoxydabilisation* and not in *dénationalisation*), one needs lots of examples for a large number of configurations. The existing databases are rather small and do not contain enough examples to carry out these studies. The only place where the needed amounts of examples could be found and collected from is the Web.

Once the data has been gathered, the linguist is faced with an even harder problem: manually checking all of them in order to remove the erroneous ones such as words in foreign languages, spelling errors, tokenization errors, etc. (see §3.2). Note that this philological verification has to be done even when the examples are collected from a standard corpus such as news archives or text databases like Frantext or the BNC. But when the examples are collected from the Web, the problem is their number. There are usually thousands of candidates which occur in millions of contexts. For some examples, one may have to go through hundreds of pages. Checking all the candidates by hand is, therefore, not practicable. Some of the collected examples have to be filtered out automatically. However, the filtering must not be too harsh because speakers are often unsure about how to spell neologisms. For example, *débogage*[2] 'debugging' is also often written *déboggage*, *débugage*, *débuggage*, etc. and the same fluctuation is observed for the corresponding verbs: *débogguer*, *débuguer*, *débugger*, etc.

2.2 Morphological aspects

In all the experiments presented here, we only look for new words that do not belong to the word lists of the common dictionaries, such as the TLFi.[3] We are also concerned only with deverbal nouns, *i.e.* derived nouns that denote the action expressed by the verb such as *production*, deverbal noun of *produce*. We are interested in this class of nouns because (*i*) they have been widely studied, (*ii*) the deverbal nouns and their verb bases share semantic features and distributional properties, (*iii*) they are coined by very productive morphological processes such as the *-age*, *-ion* and *-ment* suffixations, (*iv*) they are easy to identify and therefore easy to check, (*v*) the existing Verbaction database can be completed with our experiments, and we can use its current content for boostrapping.

French deverbal nouns can be coined by suffixation or conversion (*i.e.* non affixal derivation) such as *marcher* 'to walk' > *marche* 'a walk.' A wide range of suffixes can be used: *-age* (*nettoyer* 'clean up' > *nettoyage* 'cleaning up'); *-ion* (*organiser* 'organize' > *organisation* 'organisation'); *-ment* (*payer* 'pay' > *paiement* 'payment'); *-ade* (*ruer* 'to buck' > *ruade* 'a buck'), *-ance* (*venger* 'retaliate' > *vengeance* 'retaliation'); *-ence* (*affluer* 'flock' > *affluence* 'crowds'); *-ure* (*couper* 'to cut' > *coupure* 'a cut'), etc. Even evaluative suffixes can be used as *-ette* in *bronzer* 'suntan' > *bronzette* 'sunbath'.

The high productivity of nominalization shows up in the diversity of the registers the deverbal nouns belong to. Some of them are quite common and are just missing in the main dictionaries such as *labellisation* 'labelization'; other belong to special purpose languages as *débasage* 'debasing' (chemistry); *étrangéisation* 'make something become foreign' (philosophy); *ballonisation* 'floppy syndrome' (medicine), etc. Slang words have been also collected such as *gamellage* 'fall'.

In the following, we focus only on the nouns coined by *-age*, *-ion* and *-ment* suffixations. These nouns can therefore be searched and found on the basis of their endings: *-age*, *-ion* and *-ment* in the singular and *-ages*, *-ions* and *-ments* in the plural. However, this criterion is insufficient because of all the error sources discussed in §3.2, one of them being that many non-French nouns have these endings such as English *carriage*, *colonization* or *commitment*. One technique that can be used to find out if a word is a French deverbal noun or not is to look for contexts where it cooccurs with its possible base verb. This method

[2] *Débogage* is the term recommended by French authorities.

[3] `www.atilf.fr/tlfi.htm`

has been used by Xu and Croft (1998) in order to select morphologically related words that co-occur in a 100-words window. This kind of co-occurrence has also been studied by Baayen and Neijt (1997) who showed that the contexts where derived words occur often contain anchors used as clues for the interpretation of these words.

In the experiments we have run, the co-occurrence is looked for in the entire web page. For instance for a candidate as *débasage*, we will search for pages where it occurs with one of the following verb forms:

débasa débasai débasaient débasais débasait ... débases débasés débasez débasiez débasions débasons.

This technique is effective for two reasons: (*i*) it rejects many errors because the chances for a erroneous candidate to co-occur with a word similar but having a verb inflexional ending are quite low; (*ii*) if we suppose that documents have a good thematic and referential continuity, then the deverbal noun candidate and its base verb candidate have good chances to be semantically close.

3 Overview of the method

The experiments presented in this paper use the same method. The acquisition of the deverbal nouns and their base verbs is performed in three steps. In the first one, we look for words that are likely to be deverbal nouns. In the second one, we determine the inflected forms of their possible verb bases. In the third, we look for contexts where the deverbal noun candidates co-occur with one of these hypothetical verb forms.

3.1 A 3 steps approach

The first step of the general method is to look for words that are likely to be deverbal nouns. There are several ways to find them. When one has access to an entire index or to an entire corpus, these candidates can be identified by their endings. But when we do not have access to the index of the engine or the corpus, other techniques must be used in order to predict word forms that are likely to be deverbal nouns. The first one is to generate word forms by suffixing verb stems (*miroiter* ‘shimmer’ > *miroitage* ‘process of making a surface become sparkling’) and also stems that belong to other categories such as adjectives (*machinal* ‘mindless’ > *machinalisation* ‘act of making something become mindless’) or nouns (*mercenaire* ‘mercenary’ > *mercenairisation* ‘mercenarization’). The generation of the word forms can be done as presented in (Hathout et al., 2002) or by means of the method described in the next paragraph.

In the second step, we assume that the candidates collected in the first step are deverbal nouns and we predict the inflected forms of their verb bases. For instance, for a candidate such as *débasage*, we generate the forms listed in §2.2 by using the morphological knowledge available in Verbaction. Our method is word-based (Bybee, 1985): we have associated with every noun of Verbaction all the inflected forms of its base verb. For instance, the noun *rasage* ‘shaving’ is associated with all the forms of the verb *raser* ‘shave’. We then abstracted suffixation schemas from these couples. For instance, the couple (*rasage*, *rasons*) induces the following schemas:

```
rasage/rasons
asage/asons
sage/sons
age/ons
```

where the left-hand side represents a noun ending and the right hand side the verbal ending that has to be substituted for the former in order to get an inflected verb form. The schemas are then projected on the deverbal candidates. The inflected forms are therefore generated in one step. Because we want the prediction of the verb inflected forms to be as precise as possible, we select as model the Verbaction nouns that share the longest ending with the candidate. For instance, the model used for a candidate such as *débasage* is *rasage* and the inflected forms of its base verb (*débaser*) are generated following the example of *raser*.

In the third step, we look for attestations of the predicted inflected forms in pages which also contain the deverbal noun. A single case of such cooccurrence is enough for the noun-verb pair to be considered as valuable and submitted to manual checking: no frequency threshold is used.

3.2 Common problems and solutions

Whatever the method by which they have been harvested, candidate words come along with a lot of noise.

There is a wide litterature on error detection and correction in texts (see for example (Kukich, 1992)). However, distinguishing neologisms from errors is a specific task and processing web pages encounter specific difficulties. We identified the following noise sources and proposed some ways of dealing with them.

- *Spelling errors* are searched for with simple

Errors (%)	*-age*	*-ages*	*-ion*	*-ions*	*-ment*	*-ments*	All
Wrong part-of-speech	2.88	4.27	2.63	8.70	19.82	1.55	7.27
Tokenization error	0.82	1.71	3.95	13.83	12.78	8.53	7.35
Wrong language	3.29	6.84	5.70	5.53	24.67	31.78	11.70
Morphological error	7.00	11.11	6.14	3.95	1.32	2.33	5.01
Misc. spelling error	17.28	16.24	12.28	16.21	25.11	27.91	18.63
Correct	**68.72**	**59.83**	**69.30**	**51.78**	**16.30**	**27.91**	**50.04**

Table 1: Remaining error types for 6 deverbal noun endings

methods, for most of the genuine new words can be false positives if the correction is too greedy. Therefore we limited our algorithm (brute-force approach with a standard French dictionary) to simple editions, *i.e.* mostly to accents and repeated letters.

• *Tokenization errors* are of different types, such as extra spaces inserted in a word, or missing spaces (collided words). Both can come from the original web page, from an encoding error, or from the text conversion (especially from PDF files). We developed specific programs to detect these different situations, using both a brute-force approach and a web-based checker. More specifically, when searching for collided words, we check if an inserted space would lead to two existing words. In this case, we automatically query an online search engine to get the number of documents of the compound and split version. For example, when investigating *applaudissage* 'applauding', we examine the possibility of a missing space leading to *applaudis+sage* 'applause+wise'. The former gives 20 hits, the latter none: our conclusion is that *applaudissage* is a genuine word. On the contrary, *bulletinpage*, suspected to be a collision between *bulletin* and *page* is discarded because *bulletin+page* has 585 hits, compared to the 24 for *bulletinpage*. The same process is applied to search for extra spaces.

• *Proper names* are of no interest to us: they are discarded along with any word written in capitals.

• *Foreign language* contexts are dealt with by configuring the search engine (if any) accordingly, and by applying a stopwords-based language detection routine on the immediate context of a candidate word. However, both these methods are unsuccessful when applied to closely related languages such as Latin, Old French, Occitan, Catalan, etc. Ranaivo-Malançon (2006) studied the case of Malay and Indonesian by adding rules (based on number formats and exclusive words) to classic ngrams methods (Cavnar and Trenkle, 1994). Unfortunately, this attempt is language-specific and seems to be unfit for short contexts.

• *Computer code* is a common situation where the candidate word is in fact a variable or function name. We filter them out with the same language detection routine, as we added to our list of foreign stopwords such code-related strings as *function, var, begin*, etc. E-mail addresses and URLs are detected with simple regular expressions.

• A number of web pages are *spam documents* which can contain randomly generated strings. Although the detection of such pages is difficult, they have been more and more effectively taken into account by search engines. We nevertheless implemented a few tests, such as the detection of simple word lists (based on the fact that all words appear in the lexicographical order).

• Some candidate words belong to a *wrong part-of-speech*, such as words in *-ment* that are adverbs and not nouns (although they could be of interest in another study). Their detection would need at least some kind of automated linguistic annotation, such as part-of-speech tagging, which would be extremely ineffective in these precise situations. Dealing with unknown words when processing corpora relies on quite crude techniques, such as word-guessing, which itself relies on suffixes. POS tagging these contexts would simply lead us to consider all new *-ment* words as adverbs. Thus, this kind of error can only be solved by manually checking the contexts.

• In some cases, the base verb detection can lead to *morphological errors*. These appear when the morphological process coins the noun from something other than a verb, but which the base prediction algorithm falsely detects as such. For example, *blagounettage* 'the making of small jokes' is coined from the noun *blagounette* 'small joke', but the predicted verb *blagounetter* does not exist. Unfortunately, one of the inflected forms of this hypothetical verb is *blagounette,* thus giving a false positive because of this homography.

Overall, the filtering methods are not sufficient, and the results need to be checked manually. The breakdown of the different *remaining* error sources can be seen in table 1, for 6 different word endings. This is the result of a manual validation of 1,197 couples extracted with the third method described below (§ 5.3). As can be seen, there are important variations between suffixes. The most difficult to process is *-ment*, with only 17% precision, mostly due to the fact that this suffix is used to coin adverbs (hence the 20% POS-related error rate) and is very common in closely related languages. On the other end of the scale, *-age* and *-ion* both lead to nearly 70% precision.

It is also known that these automatic filters are overzealous, and that some correct words are discarded, but our main objective in this process is to achieve a reasonably high precision, in order to minimize manual validation.

Before presenting the actual experiments and contexts in which we used these methods, we will now take a look at the recent evolutions that led us to adapt our approach to a changing world.

4 Evolutions in using the Web as a Corpus

Corpus linguistics researchers, used to struggle to build large corpora, facing money-, time- and copyrights-related questions, realized in the early 2000s what huge, freely and easily available source of language data the web is. From that time, both technical ways to access the web and the researcher's outlooks on its use has evolved simultaneously. We briefly recall hereafter the different steps of the WAC background.

4.1 Finding a way to the wild web

Search engines (SE) came after web directories and more features have been developed while the scope of the indexed pages underwent a tremendous increase. Some engines such as Altavista, born in 1995, enabled the user to build sophisticated queries (see §5.1). Initially, the way to automate the querying of a SE was to simulate a browser's behaviour: by submitting a query with suitable parameters and parsing the results page. Year 1998 has seen the birth of Google and 5 years later, Altavista was bought twice, causing the loss of its advanced features. The SE companies started to control automated querying by developing search APIs, providing a handy way to a massive use of SE from programming languages. Nevertheless, this solution came up with some important constraints such as a maximum number of queries per day per IP.[4]

Today, whereas the search APIs are still working with previously delivered keys, no more new licenses are delivered (Google) and finding the old API is not immediate (Yahoo). The services have been replaced by products[5] intended to develop integrated web services embedded in web pages, not suitable for our task. Only Microsoft Live Search's latest API is still supported.[6] Fletcher (2007) has shown how he used it as a starting point to build a BNC-comparable corpus.

To cope with APIs restrictions and sudden changes in SE's policies, designing non-retail crawlers seems to be the ideal solution. Castillo (2004) studied how to make crawling *effective*. Among several available spiders, Heritrix is an opensource and free software, and is probably the most complete one. We will see in §5.2 that succeeding in such a scheme is a thorny issue.

4.2 The WAC initiative: from distinct goals to common challenging issues

As the practical details of the access to the web changed, the WAC problematics evolved too. Nobody wonders *"is the web a (good) corpus?"* any longer. Kilgarriff and Grefenstette (2003) already answered in the early stages and the question switched to *"is the web a corpus suitable for my task?"* The whole community usually agrees on the legitimacy of using the web. It is sometimes the only reasonable-sized source of linguistics material at disposal. The Crúbadán project (Scannell, 2007), for example, resulted in the automatic development of large text corpora for minority languages, and may not have been possible without recourse to the web.

The researchers' individual aims vary widely, from extracting large amounts of named entities to building classical general-purpose corpora. There is a also a wide range in the way they take advantage of the web. For example, Keller and Lapata (2002) use Google's result counts to retrieve frequencies of part-of-speech bigrams while Sharoff (2006) generates queries made of selected words and fetches the result pages to build large corpora. A common shared issue, apart from the way the corpus is collected and used, is the process of cleaning a messy set of pages. It has been pre-

[4] 1000 queries for Google SOAP Search API and 5000 queries for Yahoo Search API, never going beyond 1000 pages for a given query. The *per IP* restriction really matters when all workstations located behind a firewall are seen as having the same IP by the SE's server.

[5] Yahoo BOSS API and Google Ajax API.

[6] With 25000 queries per day per *application*, it is the most permissive.

sented as a tedious and unglamorous engineering task, but is a crucial bottleneck one has to deal with before using web data. The Cleaneval competition (Baroni et al., 2008) arose in year 2007 and could result in a joint effort to provide methods and tools. Unsurprisingly, even this low-level task raised non-trivial questions. Just to mention one, the task of boilerplate removal pointed out a divergence on defining what *“textual data of no linguistic interest”* means. The portion of quoted text after ’>’ in a forum post may skew statistical results of a lexicometry study whereas it may be relevant to keep it in a discourse-oriented analyse.

Our approach, confronted to these questions, is more straightforward as we do not try to build a balanced corpus, nor do we use frequency counts in any way.

5 Three different approaches

We will now present how we technically adapted our search for derived words along these years and evolutions. We will focus on our most accomplished objective, extending the Verbaction database (§2.1).

5.1 Webaffix: using AltaVista’s wildcards

The first large-scale campaign we launched (in 2001) was based on a program named Webaffix (now unfortunately obsolete), as described in (Hathout and Tanguy, 2002).

This program took advantage of the wildcard querying capability provided at this time by the Altavista search engine, which allowed for example to query for *bra*age* to get documents containing words beginning with *bra* and ending with *age*. The only restriction was that the wildcard meta-character needed to be preceded by at least 3 letters. We bypassed this constraint by building the 3000 plausible trigrams found at the beginning of French words. Another advantage of this regretted search engine was the almost unlimited query length, which allowed us to add a negative clause to the query, excluding known words from the query. A typical query would then be:

```
aqu*age -aquaplanage -aquarellage
```

(*aquaplanage* and *aquarellage* being the only two French words in our dictionary beginning with *aqu* and ending with *age*.

At this time, Altavista could be automatically queried with no restriction or quota (except for a self-imposed curtesy policy of waiting 2 seconds between queries). Each resulting web page then had to be downloaded and analysed: first to actually identify the new word candidate (no snippets were provided by Altavista), and to check for errors, as described in §3.2. This lead to the analysis of about 120,000 web pages, a process taking around 150 hours. This stage provided a list of 13,500 new nouns candidates.

Each of these words were analysed to predict their matching base verb, and thus produced 13,500 new queries, where both the candidate noun and one of its inflected base verb forms were searched for in the same document. Each resulting document was analysed to once again filter out a number of errors. As a final result, this campaign provided 1,821 new noun-verb pairs, which were finally submitted to a manual validation process, which left 926 correct ones (51%).

5.2 Trifouillette: a home-made dedicated crawler

However, these first experiments could not be continued, as Altavista stopped allowing wildcards in 2003. We then simply -and naively- decided to design our own crawler: *Trifouillette*. The principle seems pretty simple: from a given seed of URLs, fetch the pages, parse them, extract the interesting words if any, extract the links and start again.

We studied the existing crawlers but even Heritrix did not meet our needs. First, at this time, nothing was done to detect and handle spider traps.[7] Moreover, we wanted a light architecture dedicated to our task, namely not building a huge corpus, but rather gathering a collection of “interesting” pages (containing lexical creations) and storing the occurrences in a database, thus getting to the heart of the matter. This architecture enabled us to crawl and process up to 600,000 pages a day on a single machine. The NLP part of the work, though not straightforward, was usual. The pages analyser implemented the filtering heuristics described in §3.2. Conversely, the management of the crawler required unexpected daily maintenance to a discouraging extent. To spend time dealing with non-compliance with standards (servers, pages) is fair game. Cleverly handling spider traps is crafty. But using the HTTP response header to speed up the process of discarding non-French pages and discovering that all personal pages from the `free.fr` domain are assumed to be in Polish because of a misconfigura-

[7] Still today, the user manual only mentions the detection of URLs with repeated patterns or too many path segments.

	-age	*-ages*	*-ion*	*-ions*	*-ment*	*-ments*	All
Unfiltered new word forms							
Forms	48,217	12,263	158,181	38,358	71,795	11,399	340,213
Web pages	543,060	112,869	1,270,059	377,085	902,426	372,705	1,801,445
Automatic filtering							
N-V pairs	750	117	1,678	272	1,170	129	4,116
Web pages	6,862	609	17,499	2,065	28,603	5,983	53,647
Manual filtering (= estimation)*							
N-V pairs	515*	70	1,163*	141*	191*	36	2,060*
Web pages	2,954*	235	9,450*	1,733*	448*	222	14,580*

Table 2: Overview of the filtering process on Exalead Corpus

tion of the web server[8] is a bit frustrating… We also had to deal with recurrent local network dysfunctions until a new firewall made our crawler inoperative and required other modifications.

We gave up the Trifouillette project in 2006 due to a lack of time but continued to use the tools we designed as a basis for developing new specific applications.

5.3 Working with Web professionals: using Exalead's corpus

Taking advantage of a research collaboration with the Exalead company,[9] we got access in 2008 to a ready-to-use corpus of French web pages. Founded in 2000, Exalead is a software provider in Web search markets that launched in year 2006 a public search engine which indexes today 8 billion pages and is a keystone of the Quaero program.[10] The company provided us with a sample corpus made of 2.5 million pages identified as French, handling the language detection, the character encoding and the conversion into raw text. The 20GB of text pages contain 3.3 billion words, that we tokenized and indexed in a database.

Our method followed the same principles as the late Webaffix program (§5.1): we first selected word forms ending with either *-age*, *-ion* or *-ment* (or their plural counterparts) which did not appear in our French dictionary, nor in the Verbaction database. This gave us 340,213 word forms. Table 2 shows the breakdown between the 6 different word endings and the number of different web pages used to find the candidate word forms.

We then applied our filtering methods (described in §3.2), base verb prediction, and search for cooccurrence between noun and verb. This led to 4,116 new noun-verb pairs. Manual filtering on a sample of 1,197 couples by three different judges led to 599 valid pairs. The overall ratio of correct pairs is 51%, with important variations between suffixes, as explained in §3.2. Although the entire list has currently not been manually validated, it gives us a good insight at both the expected results and the general process.

First, it shows that the selected suffixes continue to provide a seemingly endless stream of new words. If our estimation is correct, the Verbaction database (currently containing 9,393 pairs) will grow by 22% with these results. Almost all new words we identified correspond to recent technical or social evolutions, as shown by these few selected examples:

- *wiitage* - *wiiter*: playing the Wii console (*i.e. wiiing*). The Wii was commercially launched in 2006.
- *sarkoïsation* - *sarkoïser*: being influenced by Nicolas Sarkozy (now French president). The word was coined by a French football player in 2006 and has been frequently used since.
- *télédéclaration* - *télédéclarer*: declaring one's income online. This has been made possible by the French tax office in 2001.
- *wambement* - *wamber*: using the social networking website Wamba (launched in 2007).

Second, it clearly shows the amount of raw data needed to extract useful information. Our estimation is that one web page out of 200 contains a new valid word pair. However, automatic filtering is quite effective in reducing the amount of data that needs to be examined manually.

6 Conclusion

As shown in these last results, we have been successfully searching for new French derived words in an ever-evolving Web. We now have the most extensive collection of French deverbal nouns available in the community. Starting 8 years ago with the opportunity to submit sophisticated queries to a compliant search engine, we tried to get along without it when it disappeared, before realising what a difficult task web-crawling is, and

[8] the pages were generated with Perl (`pl`) and the administrator probably misunderstood the role of the `Content-Language` header.

[9] `www.exalead.com`

[10] `www.quaero.org`

how it needed an industrial approach, which can only be provided by commercial search engines.

Along these different stages, our method has remained the same, our main effort being the filtering out of the erroneous contexts found in web pages. However, this evolution takes us back to a more traditional corpus approach. This has several benefits: we are less constrained in our searching (for example, the AltaVista method could not have found *wiitage*, because *wii-* is not plausible as a French word beginning), and we can now have an estimation of the huge amount of raw data necessary to get some useful linguistic material. The only visible counterpart is the bulk of data to be processed (dozens of GB and a dedicated database), while the original Webaffix program was lightweight.

This evolution also raises many methodological questions: we now are in the position to perform more sophisticated corpus linguistics inquiries on our data, such as studying more thoroughly the contexts.

Acknowledgements

The authors would like to thank Exalead for kindly letting us use their corpus and thus unburdening us of a significant part of the work.

References

R. H. Baayen and A. Neijt. 1997. Productivity in context: a case study of a Dutch suffix. *Linguistics*, 35:565–587.

R. H. Baayen, R. Piepenbrock, and L. Gulikers. 1995. The CELEX lexical database (release 2). CD-ROM. Linguistic Data Consortium, University of Pennsylvania, Pennsylvania, USA.

M. Baroni, F. Chantree, A. Kilgarriff, and S. Sharoff. 2008. Cleaneval: a Competition for Cleaning Web Pages. In *Proceedings of LREC*, Marrakech.

D. Bourigault and C. Fabre. 2000. Approche linguistique pour l'analyse linguistique de corpus. *Cahiers de Grammaire*, 25:131–151.

J. L. Bybee. 1985. *Morphology. A Study of the Relation between Meaning and Form*, volume 9. John Benjamins Publishing Company, Amsterdam.

C. Castillo. 2004. *Effective Web Crawling*. PhD Thesis, Dpt of Computer Science, University of Chile.

W. B. Cavnar and J. M. Trenkle. 1994. N-gram-based text categorization. In *Proceedings of SDAIR*, pages 161–175, Las Vegas.

W. H. Fletcher. 2007. Implementing a BNC-compareable Web Corpus. In *Building and Exploring Web Corpora, Proceedings of WAC3*, Louvain-la-Neuve.

N. Habash and B. Dorr. 2003. A categorial variation database for English. In *Proceedings of NAACL/HLT*, pages 96–102, Edmonton. ACL.

N. Hathout and L. Tanguy. 2002. Webaffix: a tool for finding and validating morphological links on the WWW. In *Proceedings of LREC*, Las Palmas.

N. Hathout, F. Namer, and G. Dal. 2002. An Experimental Constructional Database: The MorTAL Project. In Paul Boucher, editor, *Many Morphologies*, pages 178–209. Cascadilla, Somerville, Mass.

N. Hathout, M. Plénat, and L. Tanguy. 2003. Enquête sur les dérivés en *-able*. *Cahiers de grammaire*, 28:49–90.

N. Hathout, F. Montermini, and L. Tanguy. 2008. Extensive data for morphology: using the World Wide Web. *Journal of French Language Studies*, 18(1):67–85.

H. Jing and E. Tzoukerman. 1999. Information retrieval based on context distance and morphology. In *Proceedings of SIGIR*, pages 90–96, Berkeley, CA. ACM.

F. Keller and M. Lapata. 2002. Using the web to overcome data sparseness. In *Proceedings of EMNLP-02*, pages 230–237.

A. Kilgarriff and G. Grefenstette. 2003. Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29:333–347.

K. Kukich. 1992. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–439.

A. Lüdeling, S. Evert, and M. Baroni. 2007. Using Web data for linguistic purposes. In Hundt, Nesselhaut, and Biewer, editors, *Corpus Linguistics and the Web*, pages 7–24. Rodopi, Amsterdam.

M. Plénat. 2000. Quelques thèmes de recherche actuels en morphophonologie française. *Cahiers de lexicologie*, 77:27–62.

B. Ranaivo-Malançon. 2006. Automatic identification of close languages - Case study: Malay and Indonesian. *ECTI Transaction on Computer and Information Technology*, 2(2):126–133.

K. P. Scannell. 2007. The Crúbadán Project: Corpus building for under-resourced languages. In *Building and Exploring Web Corpora, Proceedings of WAC3*, Louvain-la-Neuve.

S. Sharoff. 2006. Creating general-purpose corpora using automated search engine queries. In Baroni and Bernardini, editors, *Wacky! Working Papers on the Web as Corpus*. GEDIT.

J. Xu and W. B. Croft. 1998. Corpus-based stemming using co-occurrence of word variants. *ACM Transaction on Information Systems*, 16(1):61–81.

Web Harvest of Minimal Intonational Pairs

Jonathan Howell
Dept. of Linguistics
Cornell University
jah238@cornell.edu

Mats Rooth
Dept. of Linguistics and FCI
Cornell University
mr249@cornell.edu

Abstract

This paper describes experiments on gathering spoken-language data on the web that bears on issues of the phonetics-phonology and semantics-pragmatics of intonation. The target data are tokens of fixed word strings like “than I did”, where intonation varies in a way which correlates with grammatical and pragmatic context. In a web harvest procedure, audio files were identified using a search engine based in speech-to-text, downloaded, and cut to a relevant segment under program control. In an application of such a database, an SVM classifier was trained to make a grammatically determined distinction in intonation based on purely acoustic cues. Sources of error in the retrieval are quantified.

1 Introduction

We are interested in collecting from web sources audio recordings of utterances that bear on theories of intonation. In particular, we would like to create databases of multiple repetitions of tokens embedding a fixed word string $w_1 \ldots w_n$, within which intonation varies in a way that correlates with syntax, semantics, and/or pragmatics. For instance, in comparative sentences such as (1a,b,c), there is an intuition that intonational focus in *than*-clause co-varies with the main clause in a systematic way. A generalization which turns out to be very robust (see Section 4) is that when reference varies in the subject position between the main and *than*-clauses as in (1a), the subject pronoun *I* in the *than*-clause is intonationally focused in the sense of Jackendoff (1972). When reference is constant in the subject position as in (1b) and (1c), the subject in the *than*-clause is unaccented.

1) a. She did more than I did.
b. I wish I had done more than I did.
c. I did more than I did last time.

The target sequence w_1,w_2,w_3 in this case is “than I did”. In sentences (1a-c), this substring is constant, but intonation varies in a way that correlates with the grammatical context. (1a,b) is a minimal pair, where arguably a single parameter distinguishes the clauses [than I did] in the two utterances. As articulated in theories of the semantics of focus intonation such as Rooth (1991) and Schwarzschild (1999), and accounts of the phonology-phonetics of focus intonation such as Truckenbrodt (1995) and Féry and Samek-Lodovici (2006), this is a parameter which has both a semantic/pragmatic and phonological/phonetic interpretation.

Constructing indexed web corpora in which such pairs could be retrieved, or collecting large samples of given minimal pairs from web sources, could allow both the semantic/pragmatic conditioning of the intonation and its phonetic realization to be studied and modeled on an unprecedented scale. Linguistic theories of intonation ultimately capture correlations between acoustic form and syntax, semantics and pragmatics; they make predictions about what prosodic patterns fit into what grammatical and pragmatic contexts. We would like to confront deep, logically formalized theories of this correlation with massive amounts of data harvested on the web.

This paper describes experiments in which samples for several targets were collected using a web harvest. Section 2 explains the harvest method. Section 3 evaluates the efficacy of the retrieval, discussing sources of error such as failure to retrieve an audio file over the network, and speech recognition errors. Section 4 describes an application of the data sample, where an SVM classifier was trained to make a semantically motivated distinction in the location of contrastive focus based on acoustic parameters. Section 5 gives information about additional samples being collected, and the final section offers our conclusions and suggestions about the form of web corpora of spoken language data that would be suitable for research on intonation.

2 Web harvest method

We used an external search engine with indexing based on automatic speech recognition to identify of the URLs of audio files that contain (or may contain) tokens of the target word sequence $w_1 \ldots w_n$. We aimed to use a basic approach of downloading html pages from the search engine, using simple text processing to extract URLs of audio files and other relevant information, retrieving and cutting audio files with software with a command-line interface, and using makefiles and glue languages to control the retrieval and integrate the software components.

Kohler *et al.* (2008), which discusses technology and applications for retrieval of spontaneous conversational speech, lists online search engines that index spoken language. Our survey indicated that Everyzing (search.everyzing.com) is suitable for our experiment in the following respects:

i. Searches for word strings are possible in the query language, including strings involving frequent words (stop words).
ii. Initial experimentation indicated that enough data is indexed to retrieve hundreds or thousands of tokens of the strings we are interested in.
iii. The indexed material includes a large amount of conversational data, where intonational phenomena of interest are common, and utterances are produced naturalistically.
iv. In addition to the URL of an audio file, the search engine returns time offsets for each target word. This makes it possible to automate cutting the audio files.
v. Initial experimentation indicated that, for target strings of interest, the accuracy of the engine's speech recognition was good.

Everyzing indexes both pure audio files and files with combined video and audio. Since the size of the files to be retrieved was an issue, we restricted the experiment to audio files to minimize file size. These audio files are always in mp3 format.

An experimenter first queried the engine in a browser, in order to determine whether a given string is common enough. After this, the retrieval is performed under program control, in a sequence that mimics what a human would do in interacting with the engine through a web browser.

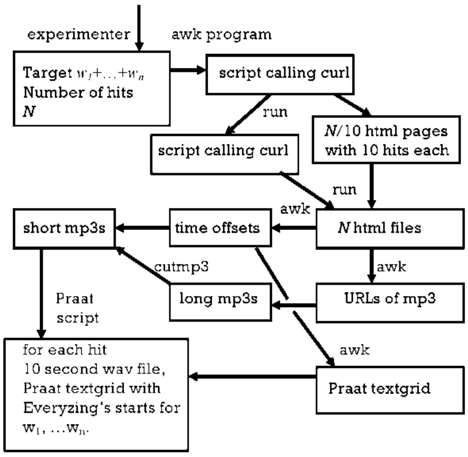


Figure 1. Workflow for mp3 retrieval and editing.

For retrieving material from the search engine, we used curl 7.16.3, which is a command line tool that retrieves data designated in URL syntax (Stenberg, 2008). The inputs to the procedure, which is diagrammed in Figure 1, are the target string and the number *N* of hits to be retrieved.

The first programmatic step constructs a shell program which contains *N*/10 calls to curl. Each involves a URL that embeds the target word string in the format "w_1+…+w_n" and an integer which functions as an index into the sequence of hits. Such a string is equivalent to the URL of the page that Everyzing displays when asked in the browser to display a group of 10 hits. Running the shell scripts retrieves *N*/10 html files, each representing 10 hits, and writes another shell script used in the next step. That script calls curl *N* times, retrieving html files for individual hits. At this point, processing with awk extracts from each file the URL of an mp3, and time offsets for the individual target words in the audio file.

Audio files are retrieved with curl, and subsequenty cutmp3, a command line program for cutting mp3 files, is used to cut a 10-second audio file from each long mp3 file, referring to the time offset (Puchalla, 2008).

Finally, we prepared data for analysis in the phonetic software package Praat (Boersma and Weenink, 2001). Mp3 files were converted to wav format, and using the time offsets of the target words, a Praat TextGrid file was prepared, which aligns the acoustic signal with the target words. Bit rate in the "than I did" dataset varied from 32 to 256 kbits/s and sampling frequency 11025 to 44100 Hz. By comparison, speech files in the often used Switchboard corpus were recorded over the telephone at 8 kbits/s and with a sample rate of 8000 Hz. Note that mp3 is a lossy

inmyopinion350.hits	html for hits 350-359
inmyopinion360.hits	html for hits 360-369
inmyopinion351.hit	html for hit 351
inmyopinion352.hit	html for hit 352
inmyopinion352.mp3name	URL of audio file
inmyopinion352.cut	time offset for hit 352
inmyopinion352.mp3	long audio file of hit 352
inmyopinion352-b.mp3	10-second audio file of hit 352

Table 1. Files from a retrieval with target "in my opinion".

compression format, which could have an impact on subsequent processing of the audio signal; however these are the available data.

In the scripts that issue requests to search.everyzing.com, we used a time delay of 25 seconds between the termination of one curl retrieval and the issuance of the next, to avoid flooding the server. We found that the audio files retrieved from various sources were often very long, and that retrieval of audio files would sometimes hang; therefore we imposed a time limit of 600 seconds for retrieving each audio file.

Files created in a retrieval run for "in my opinion" are exemplified in Table 1. The file inmyopinion352.mp3 is the full audio signal, while in inmyopinion352-b.mp3 signal has been cut to a 10-second interval flanking a putative occurrence of the target.

In the in-my-opinion run the long mp3 files had a median size of 20MB, and a maximal size of 180MB for a two hour and five minute recording of a university forum. The total size of 714 mp3s retrieved in this run is 16.4GB. The run took 24 hours.

Table 2 lists the most common domain names, indicating a predominance of radio content. WEEI, WNYC, KPBS, and WRKO are radio stations; White Rose Society is an archive of progressive radio; the items in the akamai domain comprise three AM radio stations; NPR is National Public Radio. Podtrac is site that matches podcast and advertising content.

3 Evaluation of retrieval efficacy

In a pilot experiment conducted prior to full implementation of the procedure described in Section 2, 179 purported tokens of the string "than I did" were downloaded manually by the experimenter via Everyzing and cut manually using Praat. 91 were identified as unique true occurrences of the target.

In one of several subsequent harvests using the procedure described in Section 2, 2,300 tokens of the target string "he himself" were reported by the search engine, and *N* was set at 300. The shell scripts retrieved 30 html files representing 300 hits, and then retrieved 285 individual hit html files. From these, awk generated 263 files with time-offset information (22 contained no time-offset information). 60 of the 285 mp3 files downloaded were unreadable. Upon further investigation, many of the unreadable files were in fact recoverable by a new search of Everyzing with uniquely identifying text and then manual download. This suggests corruption during the curl retrieval, rather than a corrupt file at the source.

116	a1135.g.akamai.net
110	hosted-media.podzinger.com
76	media.weei.podzinger.com
58	feeds.wnyc.org
54	media.libsyn.com
51	podcastdownload.npr.org
50	feeds.feedburner.com
39	library.kraftsportsgroup.com
33	www.whiterosesociety.org
24	www.kpbs.org
21	www.podtrac.com
21	media.wrko.podzinger.com

Table 2. The most frequent domain names in the in-my-opinion run.

An experimenter listened to all short mp3 files individually and those not containing unique occurrences of the target utterance were rejected. In 16 cases, the cut file contained inaccurate time-offsets, resulting in a short mp3 file that did not contain the purported target. Often this was due to sponsorship information in public radio podcasts which was appended to the mp3 file but did not appear in the Everyzing media player or transcription. In 25 cases, a rejected file contained an incorrectly transcribed token with a near match (e.g. *sees himself, um himself, eek himself, has himself*) or sometimes with nothing resembling the target (e.g. *building stuff, purify, independent senator*). Four of the short mp3 files were duplicates of previous files. The remaining true, unique tokens of the target numbered 154, roughly one half of the set initially queried. Other retrieval runs yielded comparable, although different results, as summarized in Figure 2.

We close this section with a comparison of the size of the datasets that can be harvested on the web with a hand-annotated speech corpus. Switchboard (Godfrey et al., 1992) contains 240 hours of speech from 2400 telephone conversations, a third of which has been made available

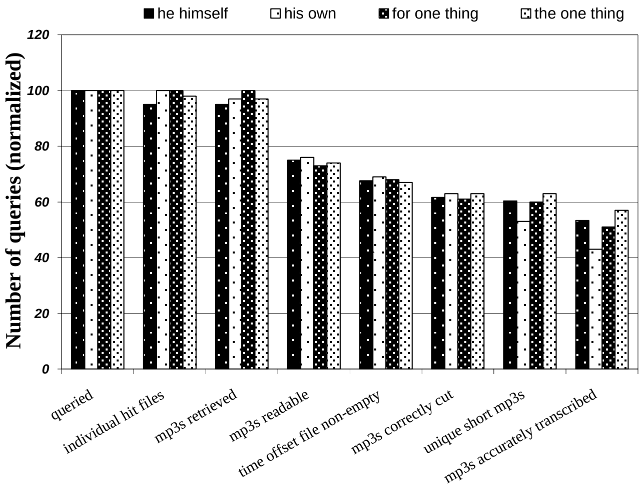


Figure 2. Detailed retrieval efficacy at different processing stages compared for 4 different retrieval runs: (normalized to 100, n=300, 100, 100, 100).

by Calhoun et al. (2005) with annotation for syntactic structure as part of the Penn Treebank (Marcus et al., 1993), dialog acts (Shriberg et al. 1998) and information status (Calhoun et al., 2005) and has formed the basis of numerous studies relating prosody, syntax and semantics (cf. Bell et al., 2009; Calhoun, 2006, 2007, 2008; Sridhar et al., 2008, Nenkova and Jurafsky, 2007; Jurafsky et al., 1998). Clearly, this type of static, richly annotated corpus offers many virtues, particularly as a standard of comparison.

Unfortunately, the restricted size of such a corpus due to the limitations of human resources means that it is not always large enough to allow statistical analysis of specific linguistic constructions. The Switchboard-1 corpus available at the Linguistic Data Consortium Online contains 26,151,602 word tokens. Figure 3 compares, for each of five targets, (a) the number of tokens contained in the Switchboard sample (b) the number of true tokens we have already collected and verified from Everyzing, and (c) the projected number of true tokens from Everyzing based on the number of hits returned and assuming a roughly 50% retrieval efficacy. While the Switchboard data may prove a useful baseline for certain target expressions, it is clear that a dynamic web harvested corpus will be not only less costly but much greater in scope. In particular, this allows us to apply machine learning techniques as an alternative to prosodic annotation by human experimenters which necessarily introduces certain theoretical assumptions such as the prosodic ontology of the Tones and Breaks Indices (TOBI) framework (Silverman et al.,1992) for prosodic annotation.

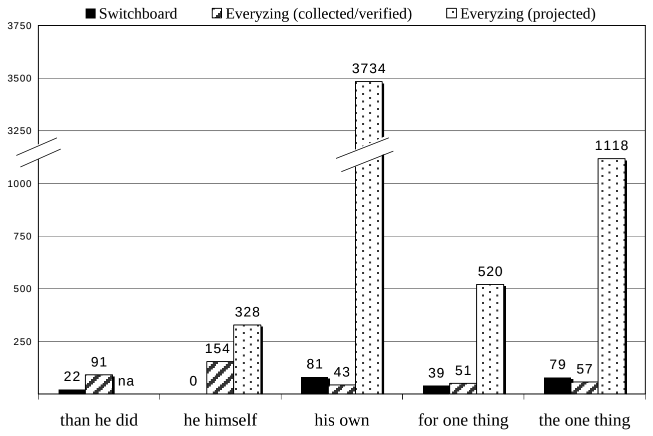


Figure 3. Comparison for each target expression of (a) number of tokens in the Switchboard corpus, (b) number of good tokens already collected and identified in the web-harvested corpus and (c) the number of projected tokens available through Everyzing at the time of harvest, based on total hit count and assuming 50% retrieval efficacy.

4 Machine learning classification

This section describes an experiment which illustrates the scientific interest of the web samples, and shows that it possible to obtain consistent results with these samples, despite variation in discourse type, recording conditions, and signal parameters, and despite the possibility of the lossy mp3 format interfering with audio processing.

On many semantic theories, unaccented material must be licensed anaphorically. In practice, however, such linguistic antecedents are not always available in the discourse; they may be inferable from the non-linguistic context.

While corpus data have the virtue of naturalness, they show extreme variation with respect to discourse context. (Laboratory-elicited data, by contrast, may be artificially controlled for discourse context although in that case the design is necessarily constrained by the experimenters' theoretical assumptions). The comparative construction discussed in Section 1 is subject to this variation, yet it has the virtue of encoding, for any given instance, an explicit antecedent. The scope of the focus (focus indicated with subscript F) is the *than*-clause, and the antecedent is contained with the main clause.

2) a. He stayed longer than [I]$_F$ did.
   antecedent: *He stayed x long*
   b. I should have liked that song a lot more than I [did]$_F$.
   antecedent: *I should have liked that song x much*
   c. I understand even less than I did [before]$_F$
   antecedent: *I understand even x little*

When the subject of the antecedent matrix clause varies from the subject of the embedded clause, theory predicts intonational prominence on *I*. When the subjects corefer, theory predicts reduced prominence. In the experiment, we trained a classifier to discriminate these two categories given only acoustic information.

As described in Section 3, we collected 179 purported tokens of the string "than I did". Each of the short sound files produced was then annotated into segments using Praat: the vowels of *than*, *I* and *did*, as well as the stop duration in *did*. Praat scripts were then used to extract 308 acoustic parameters (see Howell and Rooth, 2009).

Each token and its preceding environment was transcribed by hand. From this text, the tokens were manually classified by an experimenter into the two semantico-grammatical categories. When the subject of the main clause and the *than*-clause (i.e. *I*) varied, tokens were categorized into a class *s* (subject focus: 46/91 tokens). When the subject of the main and *than*-clauses remained constant and some contrastive postverbal material (e.g. a temporal phrase) followed (36/91 tokens) or when the subject of the main clause and *than*-clauses remained constant and no contrastive material followed (focus on *did*: 9/91), tokens were categorized into a class *ns* (non-subject focus: 45/91). This classification can be made by grammatical and semantic criteria, and is nearly uncontroversial.

Model 1: 82.4%			Model 2: 79.1%		
	predicted			predicted	
true	s	ns	true	s	ns
s	35	5	s	34	7
ns	11	40	ns	12	38
Model 3: 89.0%			Model 4: 92.3%		
	predicted			predicted	
true	s	ns	true	s	ns
s	44	8	s	43	4
ns	2	37	ns	3	41

Table 3. Contingency tables and total accuracies for predictions of different SVM classifiers using OHOCV for binary classification of subject and non-subject conditions.

A supervised support vector machine (SVM) classifier was trained in the R statistical computing environment (R Development Core Team, 2008) using an installation of the libsvm library (Chang and Lin, 2001) in package e1071 (Dimitriadou et al., 2009), using the two classes *s* and *ns*. The classifier was run with all 308 acoustic parameters (Model 1) on the 91 tokens categorized as *s* and *ns*. The success of the classifier is measured according to a one held out cross-validation (OHOCV) test. One of the 91 tokens is held out and the classifier is trained on the remaining 90. This is repeated for all of the tokens and a total accuracy is calculated on the number of successful classifications. Model 1 achieved a total accuracy of 82.4% (16 misclassifications). The results for this and following models are summarized in Table 3. A second classifier (Model 2) was tried with only 212 parameters, those extracted from *I* and *did* only, which performed marginally worse at 79.1% (19 misclassifications).

Next, we attempted different feature selection methods including a backwards-elimination technique using a random forest classifier in the R package varSelRF (Diaz-Uriarte, 2009). This produced an optimal decision tree with just a single variable: the duration of *I*. An svm classifier with just this variable (Model 3) achieved a total accuracy of 89.0% (10 misclassifications). Finally, we added to this variable the closure duration for the onset of *did*, and the difference in first and second formants at 40% of I (4*(total duration)/10) yielding a best model (Model 4) with 92.3% total accuracy (7 misclassifications).

These results offer strong empirical support of the theoretical prediction: coreference of the subject is highly correlated with reduced acoustic prominence and lack of coreference is highly correlated with increased acoustic prominence. Morover, a small set of cues for the categories involving duration and vowel quality, and not involving pitch, is sufficient to distinguish the categories acoustically.

It is not obvious that the correlation between acoustic form and semantic-grammatical context should hold up so well in such a diverse sample. We anticipate that some correlations discussed in the literature will be disconfirmed when tested against large samples harvested on the web, while others (like this one) will be confirmed and quantified.

5 Additional targets

Several other data harvests are planned or in progress. Since the machine learning classification in Section 4 revealed segmental information, in particular formant extrema, to be relevant in the detection of focus placement, we plan to harvest other targets within the same comparative paradigm, yet with different vowels: *than he did* [ij], *then they did* [ej], *than you did* [uw], *than it did* [ ]. Featural enhancement models predict that segmental features should also inform the focus placement classification for tokens with these vowels. If this is correct, one could build a successful classifier by providing information about vowel identity.

The retrieval of targets *he himself* and *his own* mentioned in Section 3 forms part of a larger harvest of targets, including other intensive reflexives, alleged to have an invariant focus pattern (e.g. Cantrall 1973; Creswell 2002; König and Gast 2006). One possible approach follows the semi-supervised method used for the comparative targets, with potentially controversial human classification into different intonational categories (e.g. *HE HIMSELF*, *he HIMSELF*). Another approach is to apply unsupervised machine learning to identify different classifications independent of human perception.

Accent type will be investigated using minimal pairs where syntax favors a particular accent. For example, most occurrences of the target *for one thing* have a “topic” accent (L*H in TOBI annotation) while most occurrences of the target *the one thing* have a “focus” accent (H*), the two predicted to differ in pitch contour. Other configurations occur with accent placement on other constituents (e.g. *except for one THING*, *that’s the one THING*). The intension is to train a classifier on these less controversial targets and then to apply it more widely to occurrences of *one thing* generally.

These targets illustrate the value of working with a very large source of data. It is possible to obtain non-trivial datasets for phenomena which, though they do not strike speakers of English as exotic, are in fact rare.

6 Discussion

We have established by example that large samples of spoken-language phenomena can be gathered on the web using simple web retrieval, text processing, and audio processing methods. The procedure is cheap. Attempted retrieval of 1000 potential tokens results in retrieval of about 750 audio files, containing hundreds of actual tokens of the target. A run of this size requires network transfer and storage of about 20GB of data. Disk capacity for this volume of data costs a few dollars. Network charge environments are readily available where transfer costs for this volume of data is on the same scale. Since the retrieval is done under program control, cost in experimenter time is also small.

The analysis in Section 3 shows that the quality of the retrieved samples varies with the target. Thinking of the system as a prototype concordance interface that presents a list of 10-second audio segments to the linguist for examination, a proportion of 50% of segments that actually contain the target seems acceptable.

It is natural to wonder whether any of the hand work in the SVM classification procedure can be automated. These steps are:

(i) Transcription of the 10-second segment.

(ii) Temporal word alignment in Praat.

(iii) Alignment of sub-phonemic acoustic events in Praat.

(iv) Classification into the semantic-grammatical categories *s* and *ns*.

Automation of any of the steps would speed up creating a dataset. Given a word transcription, there are available solutions for creating a word level alignment. For instance Yuan and Liberman (2008a,b) used a forced aligner based on the HTK HMM toolkit to create a Praat text grid with work alignments, given a word transcription. It seems likely that the same technique would be usable in (iii). This would allow the acoustic-phonetic hand work to be automated, with the additional advantage of making that work replicable.

Search.everyzing.com went offline in June 2009. Various large sites with indexing bases on speech recognition are online, such as Fox Business News and WNYC. While Google's Gaudi offering is still limited to material from the US presidential election, this could in the future be a replacement generic audio search offering.

An interesting angle is provided by individual sites that intend to expose their multimedia material to generic text search by providing transcriptions. For instance audio.weei.com (an Everyzing customer) has pages containing en embedded player for sports radio programs with functionality for search within a radio program, an mp3 download option, and a transcription. Given a list of sites, the tokens can be found with a generic text search engine, or with a textual search engine API.

The current reality is that creating datasets of sufficient size requires interacting with numerous different sites, each with its own HTML representation. Thus the text-processing work that extracts the URL of the mp3 and a time offset would have to be implemented many times, once per site. This could be compensated for by using a more sophisticated scraping technology which works with the Document Object Model representation of the page, rather than simply the string representation like the procedure in Section 3. We hope to look at available systematic solutions to this problem.

A bottleneck in the current procedure is the need for an experimenter to listen to the hits in order to select the actual tokens and create a corrected transcription of the host sentence. This is not really onerous if one is working with a few hundred examples, and at some point we want to evaluate the data as linguists anyway. But suppose 10,000 candidate tokens were available; having to listen to about 5000 incorrect tokens just to reject them would be a waste of time. We plan to look at building a targeted classifier that, for a single target, attempts to sort out the correct candidates from the incorrect ones. The classifier would be bootstrapped from a manually classified subset. This classification problem is similar to keyword spotting (e.g. Keshet et. al. 2009).

On top of general objections to basing linguistic research on commercial search engines (Kilgariff 2007), in our procedure there are sources of bias in the automatic speech recognition. It seems plausible that a speech recognizer could have substantially different recall rates for two phrase types with the same word string, but different prosodic patterns. If so, the samples collected would be biased in a way that could easily affect the evaluation of linguistic hypotheses. While it is not possible to avoid this problem within our architecture, one should try to quantify it. This might be done by finding recordings where a correct transcription is independently available. Or if working with a generic search engine, one could put test data onto the web, and measure the recall of the engine for the specific prosodic realizations of the target.

Our results and experience are suggestive about suitable forms of indexing for a web corpus of spoken language. As described in Section 3, searches for fixed word strings are useful in finding data bearing on issues on the realization and conditioning of intonation. Such searches appear to compensate for deficiencies in speech-to-text technology, because accuracy at the scale of a short tuple can be good, even if coherent transcriptions are not produced at the sentence scale. Thus it seems attractive to create web corpora of spoken language indexed by word n-grams, combined with a query system including variables and disjunctions. This would parallel web corpora and concordancing tools for written data (Fletcher, 2007).

Our results also suggest the feasibility of automatically indexing spoken-language corpora by prosodic features. Assuming that the classification results from Section 3 extend to general contexts, an SVM classifier is able to classify tokens of the first person pronoun "I" as focused or not as well as a human, based on local, paradigmatic signal features. This could make it possible to index a corpus automatically with a limited number of prosodic features.

References

Alan Bell, Jason Brenier, Michelle Gregory, Cynthia Girand, and Dan Jurafsky. 2009. Predictability effects on durations of content and function words in conversational English. *Journal of Memory and Language* 60(1):92-111.

Paul Boersma and David Weenink. 2001. Praat, a system for doing phonetics by computer. *Glot International* 5:341–345.

Sasha Calhoun. 2006. *Information Structure and the Prosodic Structure of English: a Probabilistic Relationship*. PhD thesis, University of Edinburgh.

Sasha Calhoun. 2007. Predicting focus through prominence structure. In *Proceedings of Interspeech 2007*, Antwerp, Belgium.

Sasha Calhoun. 2008. Why do we accent words? The processing of focus and prosodic Structure. Presented at *Experimental and Theoretical Advances in Prosody*, Cornell University, NY.

Sasha Calhoun, Malvina Nissim, Mark Steedman and Jason Brenier. 2005. A framework for annotating information structure in discourse. In *Frontiers in Corpus Annotation II: Pie in the Sky.* ACL2005 Conference Workshop, Ann Arbor, MI.

William R. Cantrall. 1973. Why I would relate 'own', emphatic reflexives, and intensive pronouns, my own self.' *Papers from the Ninth Regional Meeting*, eds. C. Corum, T.C. Smith-Stark and A. Weiser, 57-67. Chicago: Linguistic Society.

Chih Chang, and Chih Lin. 2001. LIBSVM: a library for support vector machines. URL http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Cassandre Creswell. 2002. The use of emphatic reflexives with NPs in English. In *Information Sharing*, eds. K. van Deemter and R. Kibble. Stanford, CA: CSLI Publications.

Ramon Diaz-Uriarte. 2009. VarSelRF: variable selection using random forests. URL http://ligarto.org/rdiaz/Software/Software.html, R package version 0.7-1.

Evgenia Dimitriadou, Kurt Hornik, Friedrich Leisch, David Meyer and Andreas Weingessel. 2009. *e1071: Misc functions of the department of statistics (e1071)*, TU Wien . R package version 1.5-19.

Caroline Féry and Vieri Samek-Lodovici. 2006. Focus projection and prosodic prominence in nested foci. *Language* 82:131-150.

William Fletcher. 2007. Implementing a BNC-Comparable Web Corpus. *Web as Corpus* 3.

John J. Godfrey, Edward Holliman and Jane McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *IEEE ICASSP-92. ACL Workshop on Discourse Annotation*.

Jonathan Howell and Mats Rooth. 2009. A corpus search methodology for focus realization. Poster presented at the 157th Meeting of the Acoustical Society of America, Portland, OR. http://hdl.handle.net/1813/13093.

Ray Jackendoff. 1972. *Semantic Interpretation in Generative Grammar*. MIT Press.

Daniel Jurafsky, Alan Bell, Eric Fosler-Lussier, Cynthia Girand, and William Raymond. 1998. Reduction of English function words in Switchboard. *Proceedings of ICSLP-98* 7.

Joseph Keshet, David Grangier, and Samy Bengio. 2009. Discriminative Keyword Spotting. *Speech Communication* 51(4):317-329.

Adam Kilgariff. 2007. Googleology is bad science. *Computational Linguistics* 33(1):147-151.

Joachim Kohler, Martha Larson, Franciska de Jong, and Wesse Kraaij. 2008. Spoken content retrieval: searching spontaneous conversational speech. *SSCS* 2008.

Ekkehard König and Volker Gast. 2006. Focused assertion of identity: A typology of intensifiers. *Linguistic Typology* 10.

Mitchell P. Marcus, Beatrice Santorini and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics* 19:313-330.

Ani Nenkova and Dan Jurafsky. 2007. Automatic detection of contrastive elements in spontaneous Speech. In *Proceedings of the IEEE workshop on Automatic Speech Recognition and Understanding* (ASRU), Kyoto, Japan.

Jochen Puchalla. 2008. Cutmp3. URL http://www.puchalla-online.de/cutmp3.html.

R Development Core Team. 2008. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria.

Mats Rooth. 1991. A Theory of Focus Interpretation. *Natural Language Semantics*, 1(1).

Roger Schwarzschild. 1999. Givenness, avoid F and other constraints on the placement of focus. *Natural Language Semantics*. 7(2):141-177.

Elizabeth Shriberg, Rebecca Bates, Andreas Stolcke, Paul Taylor, Daniel Jurafsky, Klaus Ries, Noah Coccaro, Rachel Martin, Marie Meteer, and Carol Van Ess-Dykema. 1998. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech* 41: 443-492.

Kim Silverman, Mary Beckman, John Pitrelli, Mari Ostendorf, Colin W. Wightman, Patti Price, Janet Pierrehumbert & Julia Hirschberg. 1992. A standard for labelling English prosody. *ICSLP*.

Vivek Kumar Rangarajan Sridhar, Ani Nenkova, Shrikanth Narayanan and Dan Jurafsky. 2008. Detecting prominence in conversational speech: pitch accent, givenness and focus.In *Proceedings of Speech Prosody*, Campinas, Brazil.

Daniel Stenberg. 2008. *cURL and libcurl*, http://curl.haxx.se.

Hubert Truckenbrodt. 1995. *Phonological Phrases-their Relation to Syntax, Focus, and Prominence*. PhD thesis, MIT.

Jiahong Yuan and Mark Liberman. 2008a. Speaker identification in the SCOUTUS corpus. *Journal of the Acoustical Society of America*.

Jiahong Yuan and Mark Liberman. 2008b. Vowel acoustic space in continuous speech: an example of using audio books for research. *CatCod 2008*.

Search engine based approaches for collecting domain-specific Basque-English comparable corpora from the Internet

Igor Leturia, Iñaki San Vicente, Xabier Saralegi
Elhuyar Fundazioa R&D
Zelai Haundi kalea, 3. Osinalde Industrialdea
20170 Usurbil. Basque Country
{igor, inaki, xabiers}@elhuyar.com

Abstract

In this paper we propose using search engine queries for collecting bilingual specialized comparable corpora from the Internet, an alternative to using news agencies or focused crawling which will supposedly obtain more varied corpora. The method we propose for obtaining specialized corpora on a language is based on the BootCaT method (querying search engines for random combinations of a list of seed words representative of the domain or topic and retrieving the pages returned) but, instead of the seed words, a sample mini-corpus is used as a basis for the process: most representative words are automatically extracted from it, and a final domain-filtering step is performed using document-similarity measures with this sample corpus. For obtaining the bilingual comparable corpora, two different variants of this method are proposed. One of them uses a sample mini-corpus for each language and launches the corpus-collecting processes for each language independently. The other uses only a sample mini-corpus in one of the languages, and uses dictionaries for translating the extracted seed words and performing the topic filtering for the other language. We have collected two domain-specific Basque-English comparable corpora with each of the methods, and evaluated them using corpus similarity measures.

1 Motivation

Corpora of any type are a very valuable resource for linguistic research, for language standardization and for the development of language technologies. This is more so in the case of Basque, since its standardization and normalization process begun only very recently and since language technologies for Basque are not as advanced as for other languages. But being a small language in terms of number of speakers and economic resources dedicated to it, the Basque language is not exactly rich in corpora.

So far, most of the corpora-building effort for Basque has been put into general monolingual corpora, which is completely logical, since the first step for the normalization of the language was the standardization of the general lexicon. Nowadays, although few and small compared to other languages (25 million words at most), there exist some general corpora in Basque: XX. mendeko Euskararen Corpusa[1], Ereduzko prosa gaur[2] and Klasikoen gordailua[3] are the most significant.

Now that the Academy of the Basque Language has finished with the general lexicon, and that Basque has entered universities and the labor world, there is a great need for specialized corpora, in order to normalize terminology. So far there have been two specialized corpora projects: Zientzia eta Teknologiaren Corpusa[4] (Areta *et al.*, 2007) and a project for an automatic collector of Basque specialized corpora from the Internet (Leturia *et al.*, 2008a).

Over the last years, the development of language technologies has also brought about a need for multilingual corpora, whether general or specialized, for their use in automatic terminology extraction, statistical machine translation training, etc. The Basque language has hardly any resources of this kind, except for some translation memories from public bodies,

[1] http://www.euskaracorpusa.net
[2] http://www.ehu.es/euskara-orria/euskara/ereduzkoa/
[3] http://klasikoak.armiarma.com/
[4] http://www.ztcorpusa.net

the majority of which are small and Basque-Spanish only.

However, other languages encounter this problem too, particularly for specialized areas. That is why comparable corpora are becoming increasingly popular. Although more difficult to exploit for the mentioned tasks than parallel corpora (because of their smaller alignment level, there is less explicit knowledge to extract), they are easier to obtain in large sizes, and so they also have the potential to overcome the limitations of parallel corpora, as recent research in fields such as machine translation (Munteanu and Marcu, 2005), bilingual terminology extraction (Fung and Yee, 1998; Rapp, 1999) or cross-language information retrieval (Talvensaari *et al.*, 2007) has shown. Systems that make use of this kind of corpora have also been developed for Basque (Saralegi *et al.*, 2008a; Saralegi *et al.*, 2008b). Thus the interest of an automatic tool for gathering comparable specialized corpora for Basque from the Internet.

Comparable corpora have traditionally been obtained on a supervised or directed way: out of news agencies, established research corpora (e.g. TREC or CLEF collections), by crawling certain web sites, etc. Both these approaches present some problems for our case. First, both of them need a human choice of the sources, which makes the corpora at least biased and often not very diverse. Besides, for small languages like Basque, in many domains, it would not be easy to identify good sources that would contain a significant amount of documents on the topic. Also, competition corpora do not usually include such languages. Finally, focused crawling for specialized corpora often requires domain filtering, usually based on machine learning, which needs special training for each topic, so building a generic tool for any domain is not possible. Therefore, our comparable corpora collection method is based on search engine querying.

2 Related work

2.1 Obtaining comparable corpora

Surprisingly, there is not much literature about the process of collecting comparable corpora. Most of the literature concerning comparable corpora deal with the exploitation of such resources, and simply mention that the comparable corpus has been obtained, as we have already mentioned, from news agencies (Barzilay and Lee, 2003; Munteanu and Marcu, 2005) or by crawling certain sites.

Talvensaari *et al.* (2008) do describe a system for obtaining comparable corpora based on focused crawling –the idea of focused crawling for monolingual specialized corpora was first introduced by Chakrabarti *et al.* (1999).

Some other works deal with converting comparable corpora from ‘light’ to ‘hard’ (Sheridan and Ballerini, 1996; Braschler and Schäuble, 1998; Bekavac *et al.*, 2004; Talvensaari *et al.*, 2008). The ‘light’ and ‘hard’ comparability levels for corpora were first introduced by Bekavac *et al.* (2004). A light comparable corpus would be composed of corpora from two (or more) languages composed according to the same principles (i.e. corpora parameters) which are defined by features such as domain, size, time-span, genre, gender and/or age of the authors, etc. The hard type comparability is dependent on already collected and established light comparable corpora. It is derived from them by applying certain language technology tools/techniques and/or document meta-descriptors to find out which documents in lightly comparable corpora really deal with the same or similar topic. A subset of lightly comparable corpora which has been selected by those tools/techniques, whether document-level aligned or not, can be regarded as a hard comparable corpora. Our interest, for the moment, relies on obtaining the light corpora, which again the aforementioned studies treat very superficially.

The approach most closely related to ours is that used by the BootCaT tool (Baroni and Bernardini, 2004), which introduced a new methodology for collecting monolingual domain-specific corpora from the Internet: give a list of words as input, query APIs of search engines for random combinations of these seed words and download the pages. This methodology has in some cases been used to build big general corpora (Sharoff, 2006), but for collecting smaller specialized corpora, it has become the *de facto* standard, replacing focused crawling. Although BootCaT is a monolingual corpora collector, we can expect that, by applying it to word lists on the same subject but in different languages, one could obtain light multilingual comparable corpora.

2.2 Measuring the quality of comparable corpora

The work described in this paper tries two different search engine based approaches for collecting comparable corpora from the Internet, and carries out an evaluation to see which performs best. In order to evaluate these performances, we need some way to measure the degree of comparability of a comparable corpus. However, the criteria to define comparability are not universal and depend on the type of comparable corpus we want and the task we want to use the corpus for. In our case, the comparability measure should somehow reflect domain or topic similarity and suitability for terminological extraction.

Again, the literature on this is scarce. Kilgarriff (1997) and Kilgarriff and Rose (1998) experiment with various methods for measuring corpus similarity based on word-frequency lists, and Rayson and Garside (2000) use also POS and semantic tag frequencies. But these methods are to be applied to monolingual corpora, not to multilingual comparable corpora.

Morin *et al.* (2007) suggest that, for the task of terminology extraction, the quality of a comparable corpus might be more important than its size, and show that they obtain better results with a smaller corpus if both subcorpora belong to the same register. So the genre or register could be another criterion to weigh the comparability. But word-frequency lists are not valid features for genre identification; punctuation marks and POS trigrams should be used for this task (Sharoff, 2007; Argamon *et al.*, 1998). Anyway, domain similarity is more important for terminology extraction than genre or size, so at the moment we are more interested in the former kind of comparability.

Finally, Saralegi *et al.* (2008b) propose measuring the comparability of a corpus by computing the semantic similarities at the document level. The hypothesis behind this is that the containment of many document pairs with a fairly high semantic similarity improves terminology extraction based on context similarity. So this method somehow measures the 'hardness' of 'light' comparable corpora.

3 Our approach

The aim of our research project is to develop a methodology to collect domain-specific comparable corpora from the Internet, using a search engine based approach similar to that of BootCaT. For the moment, our interest is in Basque-English corpora, but the method should work for any language pair.

The first condition, necessary but not sufficient, for two corpora to be considered domain-comparable is, obviously, that they belong to the same domain. The BootCaT tool and method can be used to obtain two such domain-specific corpora in different languages. But any loss or non-perfection in the domain-precision obtained in each of them affects the quality of the comparable corpus. The few studies that the authors have found on the topic precision obtained by BootCaT's word-list method show that this is not at all perfect (Baroni and Bernardini, 2004; Leturia *et al.*, 2008a). Thus, maximizing the domain-precision of each of the corpora obtained is one of our goals.

Then, even if both corpora were 100% domain-specific, this is not enough to guarantee a good comparability. Out of two corpora strictly on computer sciences, one could be mostly made out of texts on hardware and databases and the other on programming and networks; they could not be considered very comparable, and they would most surely not be very practical for any of the aforementioned tasks. Therefore, we are also interested in obtaining corpora as comparable as possible.

3.1 Maximizing domain precision in monolingual corpus collection

In order to try to improve the domain-precision of the BootCaT method, our approach takes, as a starting point, a sample mini-corpus of documents on the topic, instead of a list of words. This mini-corpus has two uses: first, the list of keywords to be used in the queries is automatically extracted from it; second, it is used to filter the downloaded documents according to the domain by using document-similarity techniques (Lee *et al.*, 2005).

Apart from this main contribution, we have also added some other improvements, some of them general and some others that are applied only for obtaining a better performance when the Basque language is involved.

Next we will describe the whole process we use for obtaining monolingual domain-specific corpora, which is the same as in the work of Leturia *et al.* (2008a), step by step and in more detail:

- Sample mini-corpus collection: The sample mini-corpus of documents on the target domain, which is the basis of our system, has to be collected manually. The criteria when collecting the sample is that it should be as heterogeneous as possible and cover as many different subjects of the domain as possible.

- Automatic keyword extraction: The seed words to be used in the queries are automatically extracted from the sample corpus, with the same method as used by Saralegi and Alegria (2007). The mini-corpus is automatically lemmatised and POS-tagged, and then the most significant nouns, proper nouns, adjectives, verbs, entities and multiword terms are extracted by means of Relative Frequency Ratio or RFR (Damerau, 1993) and applying an empirically determined threshold. In order to maximize the performance of the queries, the extracted list can be revised manually, to remove too specific or too local proper nouns, words that are too general and polysemous words that have other meanings in other areas.
- Querying search engines and downloading: Random combinations of the extracted seed words are sent to the APIs of search engines and the pages returned are downloaded, just as in the BootCaT method. But some changes are introduced in the method when we want a corpus in Basque, because performance of search engines for Basque is very poor, mostly due to the rich morphology of the language and to the fact that no search engine can restrict its results to pages in Basque alone. We try to solve the former by means of morphological query expansion, which consists of querying for different word forms of the lemma, obtained by morphological generation, within an OR operator. In order to maximize recall, the most frequent word forms are used, and recall is improved by up to 60% in some cases. For the latter, we use the language-filtering words method, consisting of adding the four most frequent Basque words to the queries within an AND operator, which improves language precision from 15% to over 90% (Leturia *et al.*, 2008b). These techniques are common use in IR or web-as-corpus tools for Basque (Leturia *et al.*, 2007a; Leturia *et al.*, 2007b).
- Language filter: For filtering content that is not in the target language out of bilingual documents, we use LangId, a language identifier based on character and word trigram frequencies specialized in Basque, applied at paragraph level.
- Length filter: Filtering documents by length is an effective way of reducing noise (Fletcher, 2004). In our case, we reject documents the length of which after conversion to plain text is under 1,000 characters or over 100,000 characters.
- Boilerplate removal: This is another key issue in this project, not only because boilerplate (site headers, navigation menus, copyright notices, etc.) adds noise and redundancy to corpora, but also because it can affect subsequent stages (near-duplicate detection, domain filtering, etc.). For boilerplate removal, we use Kimatu (Saralegi and Leturia, 2007), a system developed by our team that scored very well (74.3%) in the Cleaneval competition (Baroni *et al.*, 2008).
- Near-duplicates and containment detection: We have also included a near-duplicate detection module based on Broder's shingling and fingerprinting algorithm (Broder, 2000), and a containment detection method also based on Broder's works (1997).
- Domain filtering: As we have said before, we perform a final domain filtering stage. We represent both the downloaded documents and each of the documents of the sample corpus with a vector of the most significant keywords, i.e. nouns, proper nouns, adjectives and verbs. These were extracted using Eustagger, a POS-tagger for Basque (Aduriz *et al.*, 1996). The keywords are selected and weighed by some frequency measure, such as Log Likelihood Ratio or the aforementioned RFR. For measuring the similarity we use the cosine, one of the most widely used ways to measure the similarity between documents represented in the vector space model. A document is accepted in the corpus if the maximum of its cosine measures with each of the documents in the sample mini-corpus reaches an empirically defined threshold, and rejected otherwise.

3.2 Collecting multilingual corpora

With the method described above and a topic-filtering threshold that is high enough, we can obtain monolingual specialized corpora with a very high domain precision (Leturia *et al.*, 2008a). For obtaining a specialized bilingual comparable corpus, we have tried two different variants of applying this method to two different languages, which are explained below.

Different sample corpora method

The most obvious way is to use a sample mini-corpus for each language and launch the corpus collecting process independently for each of them. If the sample mini-corpora used are comparable or similar enough (ideally, a parallel corpus would be best), the corpora obtained will be comparable to some extent too (Fig. 1).

Dictionary method

The other method uses only a sample mini-corpus in one of the languages, and uses dictionaries for translating the extracted seed words (this is manually revised) and the domain-filtering vectors for the other language (Fig. 2).

This method, theoretically, presents two clear advantages: first, the sample mini-corpora are as similar as can be (it is only one), thus we can expect a greater comparability in the end; and second, we need only collect one sample corpus.

But in reality, it presents some problems too, mainly the following two: first, because dictionaries do not cover all existing terminology, we can have some OOV (Out Of Vocabulary) words and the method may not work so well –in our case, there are quite a few, although we use a combination of a general dictionary and a specialized one to maximize translation coverage –; second, we have to deal with the ambiguity derived from dictionaries, and selecting the right translation of a word is not so easy. These not at all trivial difficulties lead us to expect worse results from this method; nevertheless, we have also tried and evaluated it. To reduce the amount of OOV words, the ones that have been POS-tagged as proper nouns are included as they are in the translated lists, since most of them are named entities. And for resolving ambiguity, for the moment, we have used a naïve “first translation” approach, widely used as a baseline in NLP tasks that involve translation based on dictionaries. The basic idea this relies on is that many dictionaries order their translations according to the frequency of use.

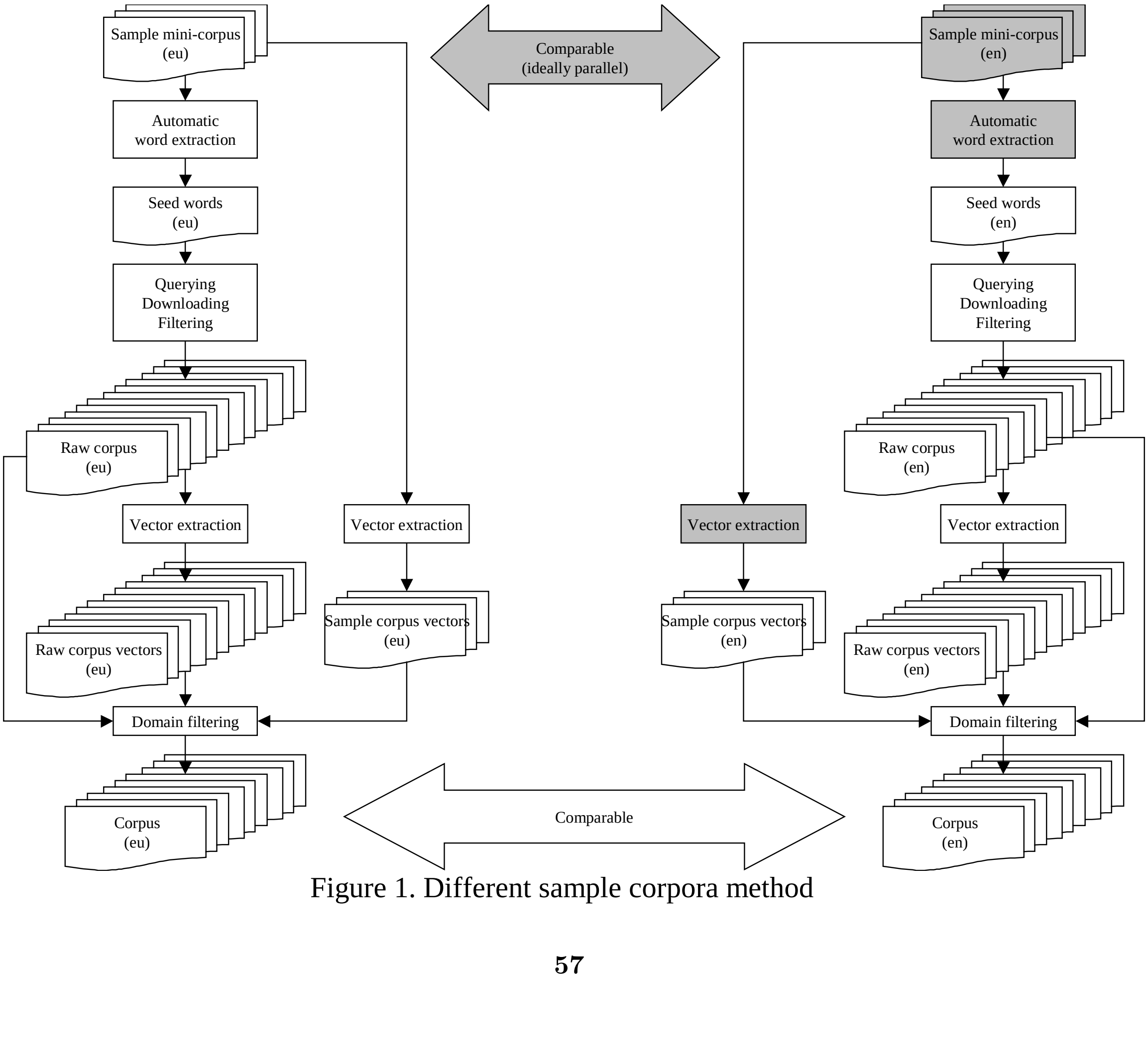


Figure 1. Different sample corpora method

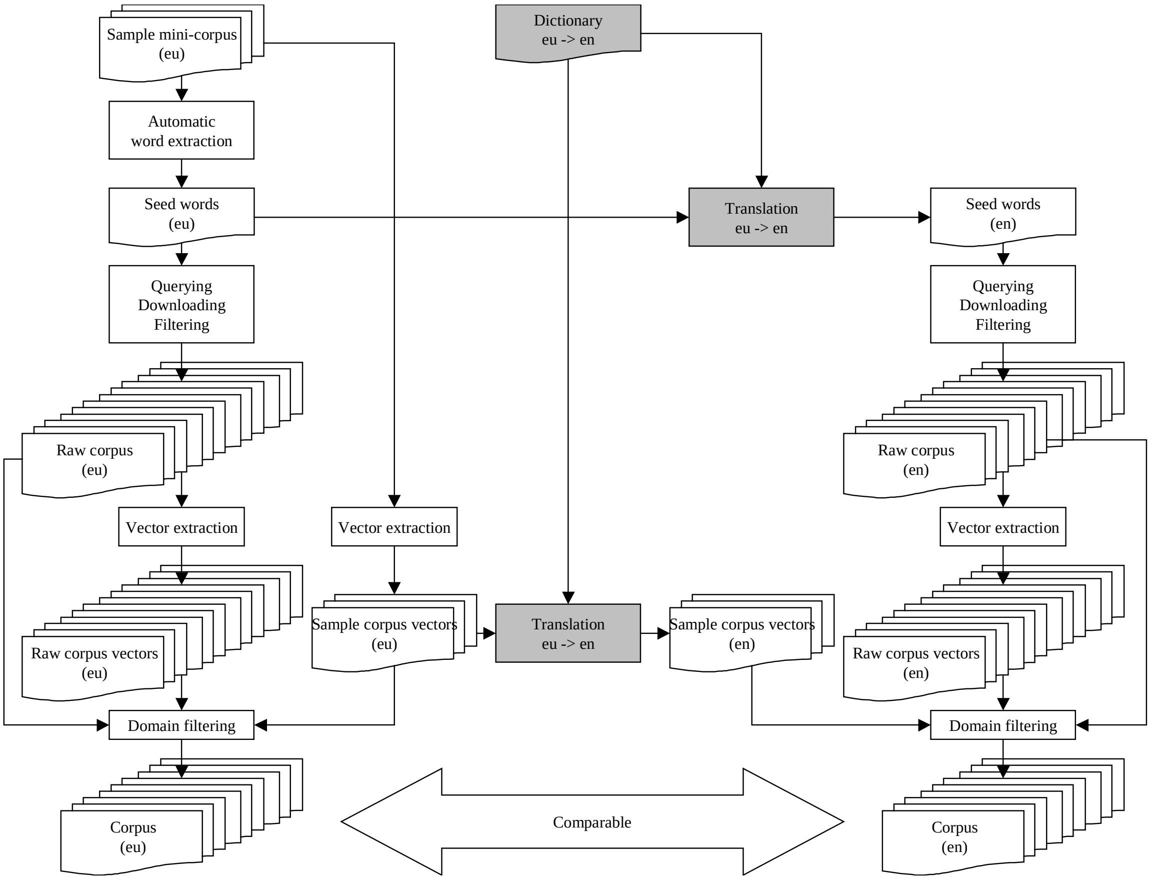


Figure 2. Dictionary method

4 Evaluation

In order to see which of the two methods obtains a higher degree of comparability, we collected two Basque-English comparable corpora, one on computer sciences and the other on tourism, with each of the two methods mentioned above. The sample mini-corpora used for computer sciences are 33 short articles (about 40,000 words) obtained from popular science magazines, and for tourism 10 short articles (about 7,000 words) obtained from tourist office websites. The English versions of the sample mini-corpora are comparable in the case of computer sciences, and parallel in the case of tourism. The final size of the computer sciences corpora amounts to 2.5 million words in each language, and in the case of tourism, 1.5 million words.

Then, for evaluating the two methods, we used two different ways to measure the comparability of the four corpora obtained: one is by calculating the cosine distance between the vectors containing all the keywords of each corpora weighted by LLR; the other is by calculating the Chi Square (χ^2) statistic for the n most frequent keywords, as described by Kilgarriff and Rose (1998). But it must be taken into account that, unlike any other corpora similarity measures mentioned in the literature, the corpora we compare are in different languages, so our measurement necessarily uses dictionaries; again, we resolve ambiguities with a first-translation approach for simplicity.

The results of the evaluation are shown in Table 1. For the cosine, higher values are better; for χ^2, a lower value indicates greater similarity. Best results are shown in bold.

Corpus	Method	Cosine, LLR, all keywords	χ^2, n most frequent keywords				
			500	1,000	5,000	50,000	All
Computer sciences	Different sample corpora	0.4102	700.61	481.57	148.70	17.60	16.55
	Dictionary	**0.4396**	**685.95**	**471.64**	**145.20**	**17.25**	**15.51**
Tourism	Different sample corpora	0.1164	382.80	**256.29**	**83.23**	**12.82**	**12.82**
	Dictionary	**0.1511**	**380.62**	261.78	86.35	13.00	13.00

Table 1. Evaluation results

5 Conclusions and future work

This paper has presented a search engine-based method for collecting specialized comparable corpora from the Internet, by obtaining two specialized, high domain-precision, monolingual corpora out of two sample mini-corpora. We tried a variant of this method that uses only one sample mini-corpus and dictionaries, to see if we could obtain similar or better comparability with less initial effort.

Although the dictionary method might *a priori* appear to be a worse method –owing to OOV words and ambiguity–, the evaluation does not confirm this. In fact, the dictionary method proved to be better in most of the measures. However, this evaluation cannot be considered conclusive, for the following reasons:

- The evaluation was done with only two corpora, which show different results for some of the measures. Besides, we now believe that tourism might not have been a good domain choice for the evaluation, because it does not completely fit into what we know as a specialized domain (interdisciplinary terminology, etc.). Evaluations with more corpora and more domains are needed before stating anything definite.
- There is not much literature on corpora similarity methods. Some measures have been proposed –mostly based on word frequency measures–, but they have not been sufficiently evaluated and indeed there is no standard measure. And regarding corpora in different languages, there is no precedent for measuring similarity. We have employed some of the proposed measures using dictionaries, and they show different results. We believe there is an urgent need for research on and standardization of multilingual corpora similarity methods.
- There might be a bias towards the dictionary method since we are using a dictionary to measure the similarity, too. To illustrate this we can imagine an extreme case, in which using the dictionary method all the seed words have been disambiguated incorrectly and the corpora obtained has nothing to see with the desired topic, but since the same dictionary and disambiguation method is applied to the keyword vectors when evaluating the similarity, the measure obtained might still be high. However, we do not see a solution for this.

For future work, we want to try to improve the dictionary-based approach; as we have already mentioned, the preliminary work needed to obtain a comparable corpus with this method is considerably reduced (only one sample mini-corpus needs to be collected); besides, there is still much room for improvement. One of the things to be tried is to see whether manual revision of the translated vectors to be used in the domain filtering yields a better performance. Another one is to try more complex translation selection techniques –instead of the first-translation approach–, and also synonymy expansion.

Furthermore, for monitoring the improvements in the methodology, we intend to make tests with more corpora and to perform further research on multilingual corpora similarity methods.

We also plan to apply the terminology extraction tool of Saralegi *et al.* (2008b) to corpora obtained with both methods and evaluate the results manually to see if our results on comparability correlate with terminological extraction tasks.

Finally, it would also be very interesting to implement a focused crawling method, download some corpora and compare the results of our method with those, to check whether the extra effort needed in focused crawling is compensated by the results.

References

Itziar Aduriz, Izaskun Aldezabal, Iñaki Alegria, Xabier Artola, Nerea Ezeiza and Ruben Urizar. 1996. EUSLEM: A Lemmatiser/Tagger for Basque. *Proceedings of EURALEX'96*, vol. I, 17-26. Euralex, Göteborg, Sweden.

Nerea Areta, Antton Gurrutxaga, Igor Leturia, Iñaki Alegria, Xabier Artola, Arantza Diaz de Ilarraza, Nerea Ezeiza and Aitor Sologaistoa. 2007. ZT Corpus: Annotation and tools for Basque corpora. *Proceedings of Corpus Linguistics 2007.* University of Birmingham, Birmingham, UK.

Shlomo Argamon, Moshe Koppel and Galit Avneri. 1998. Routing documents according to style. *Proceedings of the International workshop on Innovative Internet Information Systems (IIIS-98),* Pisa, Italy.

Marco Baroni, Francis Chantree, Adam Kilgarriff and Serge Sharoff. 2008. Cleaneval: a competition for cleaning web pages. *Proceedings of LREC 2008.* ELRA, Marrakech, Morocco.

Marco Baroni and Silvia Bernardini. 2004. BootCaT: Bootstrapping corpora and terms from the web. *Proceedings of LREC 2004*, 1313-1316. ELRA, Lisbon, Portugal.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. *Proceedings of HLT/NAACL 2003*, 16-23. NAACL, Edmonton, USA.

Božo Bekavac, Petya Osenova, Kiril Simov and Marko Tadić. 2004. Making Monolingual Corpora Comparable: a Case Study of Bulgarian & Croatian. *Proceedings of LREC 2004*, 1187-1190. ELRA, Lisbon, Portugal.

Martin Braschler and Peter Schäuble. 1998. Multilingual information retrieval based on document alignment techniques. *Proceedings of the 2nd European Conference on Research and Advanced Technology for Digital Libraries*, 183-197. Springer, Heraklion, Greece.

Andrei Z. Broder. 2000. Identifying and filtering near-duplicate documents. *Proceedings of Combinatorial Pattern Matching: 11th Annual Symposium*, 1-10. Springer, Montreal, Canada.

Andrei Z. Broder. 1997. On the resemblance and containment of documents. *Proceedings of Compression and Complexity of Sequences 1997*, 21-29. IEEE Computer Society, Los Alamitos, California, USA.

Soumen Chakrabarti, Martin van der Berg, and Byron Dom. 1999. Focused crawling: a new approach to topic-specific web resource discovery. *Proceedings of the 8th International WWW Conference*, 545-562. W3C, Toronto, Canada.

Fred J. Damerau. 1993. Generating and evaluating domain-oriented multi-word terms from texts. *Information Processing & Management*, 29:433-447.

William H. Fletcher. 2004. Making the web more useful as a source for linguistic corpora. *Corpus Linguistics in North America 2002*. Rodopi, Amsterdam, The Netherlands.

Pascale Fung and Lo Yuen Yee. 1998. An IR Approach for Translating New Words from Nonparallel, Comparable Texts. *Proceedings of COLING-ACL 1998*, 414-420. ACL, Montreal, Canada.

Adam Kilgarriff and Tony Rose. 1998. Measures for corpus similarity and homogeneity. *Proceedings of EMNLP-3*, 46-52. ACL SIGDAT, Granada, Spain.

Adam Kilgarriff. 1997. Using word frequency lists to measure corpus homogeneity and similarity between corpora. *Proceedings of workshop on very large corpora*, 231-245. ACL SIGDAT, Beijing and Hong Kong, China.

Michael D. Lee, Brandon Pincombe and Matthew Welsh. 2005. An empirical evaluation of models of text document similarity. *Proceedings of CogSci2005*, 1254-1259. Earlbaum, Stresa, Italy.

Igor Leturia, Iñaki San Vicente, Xabier Saralegi, Maddalen Lopez de Lacalle. 2008. Basque specialized corpora from the web: language-specific performance tweaks and improving topic precision. *Proceedings of the 4th Web as Corpus Workshop*, 40-46. ACL SIGWAC, Marrakech, Morocco.

Igor Leturia, Antton Gurrutxaga, Nerea areta, Eli Pociello. 2008. Analysis and performance of morphological query expansion and language-filtering words on Basque web searching. *Proceedings of LREC 2008*. ELRA, Marrakech, Morocco.

Igor Leturia, Antton Gurrutxaga, Iñaki Alegria and Aitzol Ezeiza. 2007. CorpEus, a 'web as corpus' tool designed for the agglutinative nature of Basque. *Proceedings of the 3rd Web as Corpus workshop*, 69-81. Presses Universitaires de Louvain, Louvain-la-Neuve, Belgium.

Igor Leturia, Antton Gurrutxaga, Nerea Areta, Iñaki Alegria and Aitzol Ezeiza. 2007. EusBila, a search service designed for the agglutinative nature of Basque. *Proceedings of Improving non-English web searching (iNEWS'07) workshop*, 47-54. SIGIR, Amsterdam, The Netherlands.

Emmanuel Morin, Béatrice Daille, Koichi Takeuchi and Kyo Kageura. 2007. Bilingual Terminology Mining - Using Brain, not brawn comparable corpora. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 664-671. ACL, Prague, Czech Republic.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477-504.

Reinhard Rapp. 1999. Automatic identification of word translations from unrelated English and German corpora. *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, 519-526. ACL, College Park, Maryland, USA.

Paul Rayson and Roger Garside. 2000. Comparing corpora using frequency profiling. *Proceedings of the workshop on Comparing Corpora*, 1-6. ACL, Hong Kong, China.

Xabier Saralegi and Iñaki Alegria. 2007. Similitud entre documentos multilingües de carácter científico-técnico en un entorno web. *Procesamiento del Lenguaje Natural*, 39:71-78.

Xabier Saralegi, Iñaki San Vicente and Maddalen López de Lacalle, 2008. Mining Term Translations from Domain Restricted Comparable Corpora. *Proceedings of SEPLN 2008*, 273-280. SEPLN, Madrid, Spain.

Xabier Saralegi, Iñaki San Vicente, Antton Gurrutxaga. 2008. Automatic Extraction of Bilingual Terms from Comparable Corpora in a Popular Science Domain. *Proceedings of Building and using Comparable Corpora workshop*, 27-32. ELRA, Marrakech, Morocco.

Xabier Saralegi and Igor Leturia. 2007. Kimatu, a tool for cleaning non-content text parts from HTML docs. *Proceedings of the 3rd Web as Corpus workshop*, 163-167. Presses universitaires de Louvain, Louvain-la-Neuve, Belgium.

Serge Sharoff. 2006. Creating general-purpose corpora using automated search engine queries. *WaCky! Working papers on the Web as Corpus*, 63-98. Gedit, Bologna, Italy.

Serge Sharoff. 2007. Classifying web corpora into domain and genre using automatic feature identification.. *Proceedings of the 3rd Web as Corpus Workshop*, 83-94. Presses Universitaires de Louvain, Louvain-la-Neuve, Belgium.

Páraic Sheridan and Jean Paul Ballerini. 1996. Experiments in multilingual information retrieval using the SPIDER system. *Proceedings of the 19th Annual International ACM SIGIR Conference*, 58-65. ACM, Zurich, Switzerland.

Tuomas Talvensaari, Jorma Laurikkala, Kalervo Järvelin, Martti Juhola and Heikki Keskustalo. 2007. Creating and exploiting a comparable corpus in cross-language information retrieval. *ACM Transactions on Information Systems*, 25(1):4.

Tuomas Talvensaari, Ari Pirkola, Kalervo Järvelin, Martti Juhola and Jorma Laurikkala. 2008. Focused web crawling in acquisition of comparable corpora. *Information Retrieval*, 11:427-445.

KrdWrd
Architecture for Unified Processing of Web Content

Johannes Steger
Neurbiopsychology Group
Institute of Cognitive Science
University of Osnabrück
jsteger@acm.org

Egon Stemle[*]
Computational Linguistics Group
Institute of Cognitive Science
University of Osnabrück
estemle@uos.de

Abstract

Algorithmic processing of Web content mostly works on textual contents, neglecting visual information. Annotation tools largely share this deficit as well.

We specify requirements for an architecture to overcome both problems and propose an implementation, the KrdWrd system. It uses the Gecko rendering engine for both annotation and feature extraction, providing unified data access in every processing step. Stable data storage and collaboration control scripts for group annotations of massive corpora are provided via a Web interface coupled with a HTTP proxy. A modular interface allows for linguistic and visual data feature extractor plugins.

The implementation is suitable for many tasks in the *Web as corpus* domain and beyond.

1 Introduction

Working with algorithms that rely on user-annotated Web content suffers from two major deficits:

For annotators, the presentation of Web sites in the context of annotation tools usually does not match their everyday Web experience. The lack or degeneration of non-textual context may negatively affect the annotators' performance and the learning requirements of special annotation tools may make it harder to find and motivate annotators in the first place.

Feature extraction performed on annotated Web pages, on the other hand, leaves much of the information encoded in the page unused, mainly those concerned with rendering.

In this paper, we present the design (2) and implementation (3) of the KrdWrd architecture that addresses these two issues. Section 4 contains a proof of concept in the context of CleanEval, i.e. the cleaning arbitrary web pages, and Section 5 concludes with an outlook on the possible applications and implementation improvements.

2 Design

2.1 Design Goals

We aim to provide an architecture for Web data processing based on the unified treatment of data representation and access on both the annotation and the processing side. This includes an application for users to annotate a corpus of Web pages by classifying continuous text elements and a back-end application that processes those user annotations and extracts features from Web pages for further automatic processing.

2.2 Requirements

Flexibility The system should be open enough to allow customization of every part but also, specifically provide stable interfaces for more common tasks to allow for modularization.

Stability We need a stable HTTP data source that is independent of the original Website, including any dependencies such as images, style-sheets or scripts.

Automaticity Back-end processing should run without requiring any kind of human interaction.

Replicability Computations carried out on Web page representations must be replicable across systems, including any user-side processing.

Quantity Corpus size should not influence the performance of the system and total process-

[*] Now at CIMeC, University of Trento, 38068 Rovereto.

ing time should scale linearly with the corpus.

Usability Acquisition of manually classified corpora requires a fair amount of contributions by users annotating pages. Achieving a high level of usability for the end-user therefore is paramount. As a guideline we should stay as close as possible to the everyday Web experience. We also need to provide tools for learning how to use the annotation tool and how to annotate Web pages.

2.3 Core Architecture

To address these requirements, we developed an abstract architecture, a simplified version of which is depicted in Figure 1. We outline the rationale for the basic design decisions below.

For rendering a Web page, an object tree is constructed from its HyperText Markup Language (HTML) source code. This tree can be traversed and its nodes inspected, modified, deleted and created through an API specified by the World Wide Web Consortium's (W3C) Document Object Model (DOM) Standard (Hors et al., 2004). Its most popular use case is client-side dynamic manipulation of Web pages, for visual effects and interactivity. This is most commonly done by accessing the DOM through a JavaScript interpreter. Essentially, a page's DOM tree gives access to all the information we set out to work on: structure, textual content and visual rendering data. Therefore, it serves as the sole interface between application and data.

While all browsers try to implement some part of the DOM standard (currently, Version 3 is only partially implemented in most popular browsers), they vary greatly in their level of compliance as well as their ability to cope with non-standard compliant content. This leads to structural and visual differences between different browsers rendering the same Web page.

Therefore, to guarantee *replicability*, we require the same DOM engine to be used through the envisioned system.

To reach a maximal level of *automaticity* and not to limit the *quantity* of the data, it is important that data analysis takes place in a parallel fashion and does not require any kind of graphical interface, so it can e.g. be executed on server farms. On the other hand we also need to be able to present pages within a browser to allow for user annotation. Consequently, the same DOM engine needs to power a browser as well as a headless back-end application, with *usability* being an important factor in the choice of a particular browser.

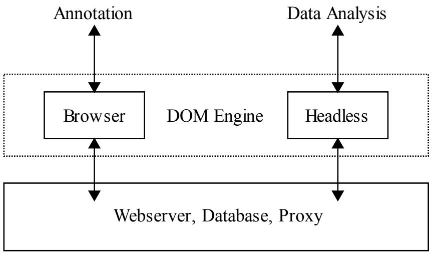


Figure 1: Basic KrdWrd Architecture: both users annotating corpus pages through their Web browser and back-end applications working on the data run the same DOM engine. The central server delivers and stores annotation data and coordinates user submissions.

The annotation process, especially the sequence of presentation of pages, is controlled by a central Web server – users cannot influence the pages they are served for annotation. Thereby any number of concurrently active users can be coordinated in their efforts and submissions distributed equally across corpus pages. All data, pristine and annotated, is stored in a database attached to the Web server. This setup allows the architecture to scale *automatically* with user numbers under any usage pattern and with reasonable submission *quantities*.

Stability of data sources is a major problem when dealing with Web data. As we work on Web pages and the elements contained in them, simple HTML dumping is not an option – all applications claiming to offer full rewriting of in-line elements fail in one way ore another, especially on more dynamic Web sites. Instead, we use a HTTP proxy to cache Web data used in our own storage. By setting the server to grab content only upon first request and providing an option to turn off download of new data, we can create a closed system that does not change once populated.

3 Implementation

We maintain the implementation in a source code repository at `http://krdwrd.org`. The documentation includes pointers to the required external software.

This section will first describe the DOM engine and its use by browser and back-end application (3.1), then the details of the implementation of central storage and control (3.2), and will end with listing possible feature extractors for the back-end (3.3).

3.1 DOM Engine

The choice of DOM engine is central to the implementation. We reviewed all major engines available today with respect to the requirements listed in 2:

The KDE Project's KHTML drives the Konquerer browser and some more exotic ones, but lacks a generic multi-platform build process.

This practical limitation is lifted by Apple's fork of KHTML, called WebKit. It is the underlying engine of Safari browsers on Mac OS X and Windows. There also exists a Qt and a GTK based open source implementation. Whereas they are quite immature at the moment and not very widely used, this will change in the future and WebKit will certainly become a valuable option at some point.

Whereas the open source variant of Google's browser, *Chromium*, promises superior execution speed by coupling WebKit with its own V8 JavaScript engine, it suffers from the same problem as WebKit itself namely, not being stable enough to serve as reliable platform – the Linux client for example is barely usable, a Mac client does not even exist, yet.

We also briefly evaluated Presto (Opera) and Trident (Microsoft), but discarded them due to their proprietary nature and lack of suitable APIs.

The Gecko engine (Mozilla Corporation), in conjunction with its JavaScript implementation Spidermonkey, marks a special case: It implements XUL (Goodger et al., 2001), the XML User Interface Language, as a way to create feature rich cross-platform applications. The most prominent of those is the Firefox browser, but also e.g. Thunderbird, Sunbird and Flock are built with XUL. An add-on system is provided that allows extending the functionality of XUL applications to third-party code, which gains full access to the DOM representation, including the XUL part itself. The proposed KrdWrd back-end can be implemented in the same manner as Firefox: provide custom JavaScript and XUL code on top of Mozilla's core XUL Runner. Code can easily be shared between a browser add-on and XUL applications and unsupervised operation is trivial to implement in a XUL program.

Given the synergy attainable in the XUL approach and Firefox' popularity amongst users, it was a simple decision to go with Mozilla Gecko for the core DOM implementation. We note that WebKit's rise and fast pace of development might change that picture in the future.

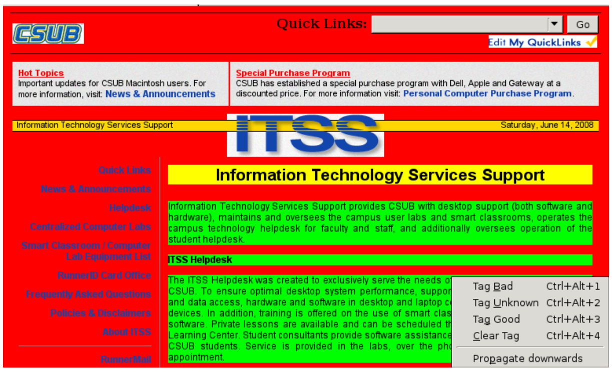
3.1.1 Firefox Add-on

Interactive visual annotation of corpus pages via Web browser is realized by the KrdWrd Firefox Add-on. The imposed annotation *base data* (Müller and Strube, 2003) are text elements in the DOM tree, which are non-overlapping word-, phrase-, and character-level strings, i.e. we do not superimpose a different structure. [1] The annotation then, is non-hierarchical, i.e. a single node can only be classified into one class at a time, and continuous, i.e. a class can only be assigned to one node at a time.

To facilitate adoption, it comes with a comprehensive user manual and an interactive tutorial (see below in 3.2.1). For easy setup, Firefox's proxy configuration is automatically pointed to a preconfigured host, respective credentials are auto-added to the password manager and the user is directed to a special landing page upon successful installation. The proxy feature also serves as a nice example of code shared between add-on and application. Furthermore, the installation binary is digitally signed, so the user does not have to go through various exception dialogs.

Once installed, the functionality of the Add-on is available via a broom icon in the status bar. Whereas it offers several functions centered around annotation and corpus selection, its core feature is simple: In highlight mode (the broom turns fuchsia) the mouse hovering over the page will highlight the text blocks below the cursor. The block can then be annotated using the context-menu or a keyboard short-cut, which will change its color to the one corresponding to the annotation class. Figure 2 shows a fully annotated page and the context-menu.

[1] However, while grabbing documents we surround text nodes of running text with additional <KW>-Elements: this delimits large amounts of text under a single node in the DOM tree, i.e. when the whole text could only be selected as a whole, these elements loosen this restriction but, on the other hand, do not affect the rendering of the Web page or other processing steps.



Hot Topics
Important updates for CSUB Macintosh users. For more information, visit: News & Announcements

Special Purchase Program
CSUB has established a special purchase program with Dell, Apple and Gateway at a discounted price. For more information visit: Personal Computer Purchase Program.

Information Technology Services Support Saturday, June 14, 2008

Information Technology Services Support

Information Technology Services Support provides CSUB with desktop support (both software and hardware), maintains and oversees the campus user labs and smart classrooms, operates the campus technology helpdesk for faculty and staff, and additionally oversees operation of the student helpdesk.

ITSS Helpdesk

The ITSS Helpdesk was created to exclusively serve the needs o
CSUB. To ensure optimal desktop system performance, suppor
and data access, hardware and software in desktop and laptop c
devices. In addition, training is offered on the use of smart clas
software. Private lessons are available and can be scheduled th
Learning Center. Student consultants provide software assistance
CSUB students. Service is provided in the labs, over the pho
appointment.

Figure 2: Web pages can be annotated with the KrdWrd Firefox Add-on by hovering over the text by mouse and setting class labels by keyboard short-cut or pop-up menu.

3.1.2 XUL Application

The XUL application consists of a thin JavaScript layer on top of Mozilla's XUL Runner. It mainly uses the XUL browser control to load and render Web pages and hooks into its event handlers to catch completed page load events and the-like. Without greater C level patching, XUL still needs to create a window for all of its features to work. In server applications, we suggest using a virtual display such as Xvfb to fulfill this requirement.

During operation the application parses the given command-line arguments, which triggers the loading of supplied URLs (local or remote) in dedicated browser widgets. When the "load complete" event fires, one of several extraction routines is run and results are written back to disk. The implemented extraction routines are:

grab for simple HTML dumps and screen-shots,

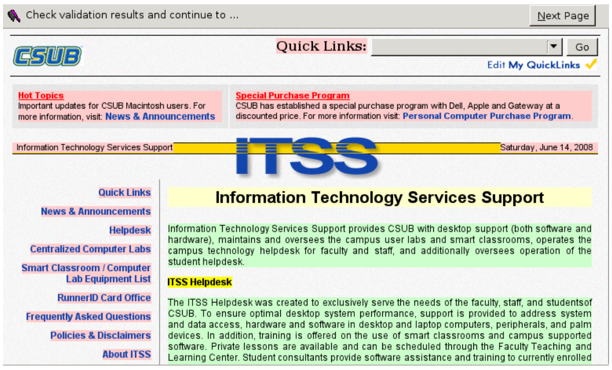
diff for computing a visual difference rendering of two annotation vectors for the same page,

merge for merging different annotations on the same Web page into one in a simple voting scheme, and

pipe for textual, structural and visual data for the feature pipelines.

3.2 Storage and Control

Central storage of Web pages and annotation data is provided by a database. Clients access it via CGI scripts executed by a Web server while the back-end uses python wrapper scripts for data exchange.



Check validation results and continue to ...

Hot Topics
Important updates for CSUB Macintosh users. For more information, visit: News & Announcements

Special Purchase Program
CSUB has established a special purchase program with Dell, Apple and Gateway at a discounted price. For more information visit: Personal Computer Purchase Program.

Information Technology Services Support Saturday, June 14, 2008

Information Technology Services Support

Information Technology Services Support provides CSUB with desktop support (both software and hardware), maintains and oversees the campus user labs and smart classrooms, operates the campus technology helpdesk for faculty and staff, and additionally oversees operation of the student helpdesk.

ITSS Helpdesk

The ITSS Helpdesk was created to exclusively serve the needs of the faculty, staff, and studentsof CSUB. To ensure optimal desktop system performance, support is provided to address system and data access, hardware and software in desktop and laptop computers, peripherals, and palm devices. In addition, training is offered on the use of smart classrooms and campus supported software. Private lessons are available and can be scheduled through the Faculty Teaching and Learning Center. Student consultants provide software assistance and training to currently enrolled

Figure 3: During the tutorial, a Visual Diff between the user's submission and the sample data is presented right after submission. Here, the annotation from Figure 2 was wrong in tagging the sub-heading "ITSS Helpdesk": the correct annotation (*yellow*) is highlighted in the feedback in dark color – contrary to the heading "Information Technology Services Support" that was tagged correctly and hence, shows up in light color.

3.2.1 Web Server

Server-side logic is implemented by Python CGI scripts, thus any Web server capable of serving static files and executing CGI scripts is supported. Users can access the server directly by URL or via the Firefox Add-on menu. An overview page rendered by the server provides a submission overview as well as a detailed per-corpus submission list. In conjunction with the Add-on, server side scripts control serving of corpus pages by summing over submissions in the database and randomly selecting a page from those with the least total submission number. The Web server also delivers the actual HTML data to the client, whereas any embedded objects are served by the separate proxy server. Furthermore, it controls the tutorial: Users are presented with sample pages and asked to annotate them. Upon submission, a server side script compares the user's annotation with a reference annotation stored in the database and generates a page that highlights differences. The result is delivered back to the user's browser, as seen in Figure 3.

3.2.2 Database

The database mainly stores the raw HTML code of the corpus pages. User submissions are vectors of annotation classes, the same length as the number of text nodes in a page. In addition there is a user mapping table that links internal user ids to external authentication. Thereby user submissions are anonymized, yet trackable by id.

Given the simple structure of the database model, we choose to use zero-conf database backend *sqlite*. This should scale up to some thousand corpus pages and users.

It is important to note that any database content must be pre-processed to be encoded in UTF-8 only. Unifying this bit of data representation at the very start is essential to avoid *encoding hell* later in the process. To this end, we rely on Mozilla's *Universal Charset Detector*[2], which is part of the Gecko engine, a mature *composite approach to language/encoding detection* (Li and Momoi, 2001) – the UTF-8 encoded output is fed into the database.

3.2.3 Proxy

Any object contained in the corpus pages needs to be stored and made available to viewers of the page without relying on the original Internet source.

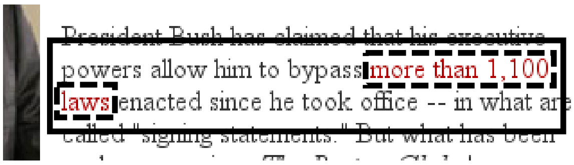
Given an URL list, initial population of the proxy data can easily be achieved by running the XUL application in grabbing mode while letting the proxy fetch external data. Afterwards, it can be switched to block that access, essentially creating a closed system. We found WWWOffle to be a suitable proxy with support for those features while still being easy to setup and maintain.

3.3 Feature Extractors

The XUL Application extracts information from corpus pages and dumps it into the file-system, to serve as input to specialized feature extractors. This implementation focuses on feature extraction on those nodes carrying textual content, providing one feature vector per such node. We therefore generate one feature vector per such node through a linguistic, visual and DOM-tree focused pipeline.

3.3.1 Text

For linguistic processing, the Application dumps raw text from the individual text nodes, with leading and trailing whitespace removed, converted to UTF-8 where applicable, i.e. the quirks of handling languages such as Chinese and Japanese, or even bi-directional languages like Hebrew are transparent to our processing and the subsequent applications need to be capable of handling these languages. External applications can read these data and write back the feature vector resulting from their computation in the same format.



President Bush has claimed that his executive
powers allow him to bypass more than 1,100
laws enacted since he took office -- in what are
called "signing statements." But what has been

Figure 4: Coordinates of a node's bounding box (straight) and text constituents (dotted) as provided to the visual processing pipeline.

For Computational Linguistic tools relying on phrase-level structured input, e.g. tokenizers, the Application can also dump raw text that more closely resembles the rendered output, i.e. paragraphs, spanning multiple nodes, are merged together and dumped in one line; each line – and hence, feature vector – is then duplicated as many times as nodes that are spanned.

3.3.2 Structural

During an Application run, a set of "DOM-Features" is directly generated and dumped as feature vector.

Choosing the right DOM properties and applying the right scaling is a non-trivial per-application decision. Our reference implementation includes features such as depth in the DOM-tree, number of neighboring nodes, ratio text characters to HTML code characters, and some generic document properties as number of links, images, embedded objects and anchors. We also provide a list of the types of node preceding the current node in the DOM-tree.

3.3.3 Visual

For visual analysis, the Application provides full-document screen-shots and coordinates of the bounding rectangles of all text nodes.[3] When text is not rendered in one straight line, multiple bounding boxes are provided as seen in Figure 4. This input can be processed by any application suitable for visual feature extraction.

For simple statistics dealing with the coordinates of the bounding boxes, we use a Python script to generate basic features such as total area

[2] http://www.mozilla.org/projects/intl/detectorsrc.html

[3] This Extractor requires at least XUL Runner Version 1.9.2 (corresponding to Firefox Version > 3.5) which is still in beta at the time of this writing.

Table 1: BootCaT seed terms for *Canola* corpus

history	coffee	salt
spices	trade road	toll
metal	silk	patrician
pirate	goods	merchant

covered in pixel, number of text constituents, their variance in x-coordinates, average height and the-like.

4 Case Study

The current implementation comprises an extensive system for pre-processing and automated cleaning of Web pages, i.e. a typical Web-as-corpus task, where users are provided with accurate Web page presentations and annotation utilities in a typical browsing environment, while supervised machine learning algorithms also operate on representations of the visual rendering of Web pages.

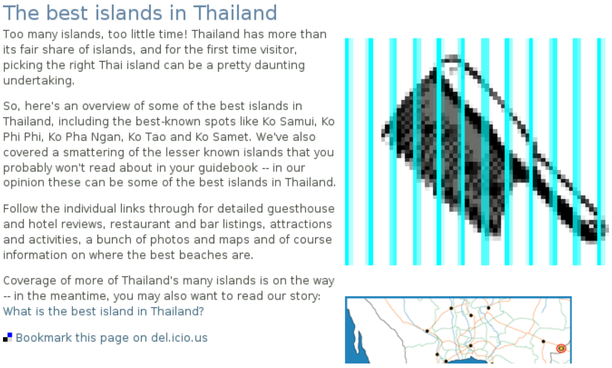
The sequence of steps includes corpus creation and acquisition of hand-annotated training data on that corpus (4.1), feature extraction (4.2), training of a classifier and producing annotated test results (4.3).

The underlying data, tools, and programs are bundled with the KrdWrd distribution as usage example.

4.1 Data Acquisition

Gathering a set of sample pages is the first step before utilizing people to tag new data. Therefore, we acquired a new corpus named *Canola* by using the BootCaT (Baroni and Bernardini, 2004) tool to produce a URL list from the seed terms in Table 1 using the Yahoo search engine.

To populate the proxy, we ran the Application on every URL once and also extracted the textual content of the pages. We then filtered for text lengths between 500 and 6,000 characters [4] and ran the Application once again, this time dumping the raw HTML code of the pages in UTF-8 format. During this second pass, the proxy is switched to block access to external sources. This ensures that no dynamic external content makes it into the corpus data, while letting innocent content pass. See Figure 5 for an example.



The best islands in Thailand

Too many islands, too little time! Thailand has more than its fair share of islands, and for the first time visitor, picking the right Thai island can be a pretty daunting undertaking.

So, here's an overview of some of the best islands in Thailand, including the best-known spots like Ko Samui, Ko Phi Phi, Ko Pha Ngan, Ko Tao and Ko Samet. We've also covered a smattering of the lesser known islands that you probably won't read about in your guidebook -- in our opinion these can be some of the best islands in Thailand.

Follow the individual links through for detailed guesthouse and hotel reviews, restaurant and bar listings, attractions and activities, a bunch of photos and maps and of course information on where the best beaches are.

Coverage of more of Thailand's many islands is on the way -- in the meantime, you may also want to read our story: What is the best island in Thailand?

Bookmark this page on del.icio.us

Figure 5: IFrames with dynamic URLs which usually come from advertisements are blocked as a nice side-effect of the Proxy setup.

The resulting HTML is post-processed to ensure that references and encodings are consistent: The head tag is expanded by a `<base href="`*`original url`*`" />` line, so a browser later viewing the dumped HTML will request embedded objects by their original URLs, which can then be served by the proxy. After removing any non-UTF-8 encoding hints, the data is fed into the database's page table, with a unique page id and the corpus id.

The pre-processed data is now ready to be processed by annotators. For gathering training data, students were asked to go through the ten Web page annotation tutorial once – to get acquainted with the annotation tool, i.e. the Add-on, and different aspects of how to apply the guidelines [5] to real-world Web pages – and then annotate pages from the *Canola* corpus as part of an homework assignment. The annotation process consisted of tagging text on Web pages with *three* tags 'good', 'bad', and 'uncertain'.

Over the course of two weeks, about 60 students provided a total average of 7.75 annotations per page. As the time data in Figure 6 suggests, users learn quickly; Average per-page annotation times drop well below three minutes after some training. The tutorial with its ten pages took on average 22 minutes to complete; note however, these pages were shortened and stripped down to illustrate particular aspects of Web pages.

Integration of the Add-on in users' environments was flawless and we did not receive any reports of usability or general handling problems.

[4] ...for Chinese these numbers had to be cut down to 50 and 600, however.

[5] A refined version of the official 'CLEANEVAL: Guidelines for annotators' `http://cleaneval.sigwac.org.uk/annotation_guidelines.html` available at `https://krdwrd.org/manual/html`.

Manual inspection of submissions also did not show any anomalies, but to the contrary, indicated that submitters took great care to provide adequate annotations (c.f. Figure 7).

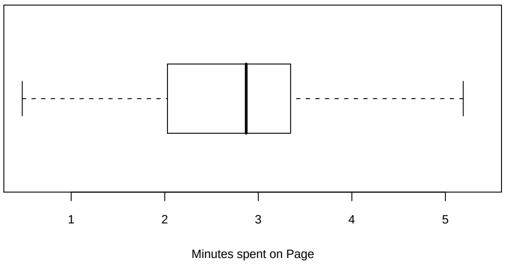


Figure 6: Time spent for annotation of a single Web page across all annotators of the *Canola* corpus.

The data obtained from user annotations was next merged into a single corpus using the Application's *merge* function (c.f. 3.1.2), resulting in a total of 216 corpus pages, each backed by up to 8 user submissions. Different treatment of JavaScript on the client side resulted in partial misalignment on some pages: dynamic client code had inserted or re-ordered nodes in some instance while not in others. We extended the merge procedure to accept some fuzziness in node matching, but still lost data from about 5% of submissions that could not be re-aligned. Until this problem is solved, we turn off JavaScript for Web content via the Firefox Add-On. Note that attaching unique IDs to text nodes is only a partial solution to this problem: A common JavaScript idiom is to clone an existing element and to populate it with new content, ultimately leading to different nodes with the same "unique" ID.

4.2 Extraction Pipeline

Feature Extraction commences by running the KrdWrd application extraction pipeline over the merged data obtained during annotation. For the *Canola* corpus' 216 pages, it took 2.5 seconds on average per page to generate text (2.5 million characters total), DOM information (46575 nodes total), screen-shots (avg. size 997x4652 pixels) and a file with the annotation target class for each text node.

We only used the stock KrdWrd features on the DOM tree and visual pipeline. For computing textual features, we borrowed Victor's (Spousta et al., 2008) text feature extractor.

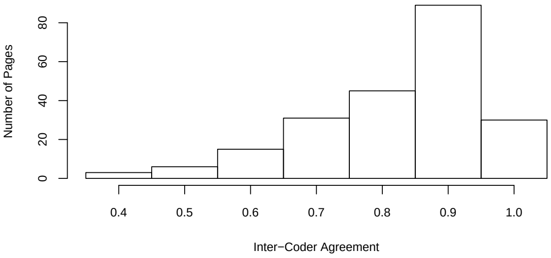


Figure 7: Fleiss's multi-π agreement (Artstein and Poesio, 2008) between submissions for pages over the *Canola* corpus.

4.3 Experiment

We used the data gathered by the feature extraction for training a Support Vector Machine (Chang and Lin, 2001). We used an RBF kernel with optimal parameters determined by a simple grid search to create ad-hoc models on a per-pipeline basis. The total number of feature vectors corresponded to the number of text nodes in the corpus and was 46575. Vector lengths for the different pipelines and test results from 10-fold cross validation are shown in Table 2.

Although the results for the single pipelines look quite promising – especially the surprisingly good performance of the visual pipeline given its limited input – combinations of feature sets in a single SVM model perform only marginally better. We therefore suggest running separate classifiers on the feature sets and only merging their results later, possibly in a weighted voting scheme. DOM features would certainly benefit most from e.g. a classifier that can work on structured data.

4.4 Inspecting Classifier Results

The classification results can be back-projected into the DOM-trees using the Application's *diff* function. As in the tutorial for annotators, it produces a visual diff, showing where the classifier failed. Note that these results are just Web pages, so they can be viewed anywhere without the help of the Add-on. This quickly turned out to be a valuable tool for evaluation of classification results.

Table 2: 10-fold cross validated classification test results for different combinations of the textual (cl), DOM-property based (dom) and visual (viz) pipelines on the *Canola* data set obtained using stock SVM regression with a RBF kernel.

Modules	Feat.	Acc.	Prec.	Recall
cl	21	86%	61%	76%
dom *	13	65%	64%	56%
viz *	8	86%	64%	82%
cl dom *	34	67%	74%	57%
dom viz *	21	67%	72%	59%
cl viz	29	86%	63%	78%
cl dom viz	42	68%	76%	58%

* data obtained by training on reduced number of input vectors.

5 Conclusion

Employing KrdWrd in the *Canola* case study showed that we achieved what we set out for and gave some valuable experience for possible improvements:

The KrdWrd Firefox Add-On is the first tool for Web page annotation that integrates flawlessly into a users daily browsing experience. It is unobtrusive and has a simple and intuitive user interface. Users quickly learn how to annotate and produce quite uniform results, given sufficient annotation guidelines.

The KrdWrd application and supporting infrastructure are a reliable platform under a real-world usage scenario. By decoding any input data to UTF-8 at the moment it enters the system and ensuring that we explicitly deliver UTF-8 exclusively throughout the system, we circumvented all usual encoding problems.

The overall handling of JavaScript is not satisfactory. To address the diversions between submits occurring after dynamic client-side JavaScript execution on different clients, the Add-on could hook into the node creation and clone processes. They could be suppressed entirely or newly created nodes could grow a special id tag to help identifying them later.

For result analysis, we would like to expand the visual diff generated from classification results. Showing results from separate runs on different subsets of the data or different parameters on one page would facilitate manual data inspection. Presenting selected feature values per node might also help in developing new feature extractors, especially in the DOM context.

Furthermore, we would like to integrate the JAMF framework (Steger et al., 2008), a component-based client/server system for building and simulating visual attention models, into the tool chain. This would allow for features based on the analysis of the rendered pages akin to how humans perceive these pages while browsing.

Summarizing, we designed and implemented an architecture for holistic treatment of Web pages in classification tasks. We demonstrated that the KrdWrd system can be used to automatically build an annotated corpus from user submissions. We also showed the broad set of features for text, structure and imagery it can help to extract, and how their contribution to classification can be assessed graphically.

References

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Marco Baroni and Silvia Bernardini. 2004. Bootcat: Bootstrapping corpora and terms from the web. Proceedings of LREC 2004.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Ben Goodger, Ian Hickson, David Hyatt, and Chris Waterson. 2001. Xml user interface language (xul) 1.0. Recommendation, Mozilla.org.

Arnaud Le Hors, Philippe Le Hgaret, Lauren Wood, Gavin Nicol, Jonathan Robie, Mike Champion, and Steve Byrne. 2004. Document object model (dom) level 3 core specification. Recommendation, W3C.

Shanjian Li and Katsuhiko Momoi. 2001. A composite approach to language/encoding detection. In *19th International Unicode Conference*.

Christoph Müller and Michael Strube. 2003. Multi-level annotation in mmax. In *Proc. of the 4th SIGDIAL*.

Miroslav Spousta, Michal Marek, and Pavel Pecina. 2008. Victor: the web-page cleaning tool. In *Proceedings of the 4th Web as Corpus Workshop (WAC4) – Can we beat Google?*

Johannes Steger, Niklas Wilming, Felix Wolfsteller, Nicolas Höning, and Peter König. 2008. The jamf attention modelling framework. In Lucas Paletta and John K. Tsotsos, editors, *WAPCV*, volume 5395 of *Lecture Notes in Computer Science*, pages 153–165. Springer.

Examining the Use of Region Web Counts for ESL Error Detection

Joel R. Tetreault
Educational Testing Service
660 Rosedale Road
Princeton, NJ, USA
JTetreault@ets.org

Martin Chodorow
Hunter College of CUNY
695 Park Avenue
New York, NY, USA
martin.chodorow@hunter.cuny.edu

Abstract

Significant work is being done to develop NLP systems that can detect writing errors produced by non-native English speakers. A major issue, however, is the lack of available error-annotated training data needed to build statistical models that drive these major systems. As a result, many systems are trained on well-formed text with no modeling of typical errors that non-native speakers produce. To address this issue, we propose a novel method of using geographic region-specific web counts to detect typical errors in the writing of non-native speakers. In this paper we describe the approach, and present an analysis of the issues involved when using web counts.

1 Introduction

In recent years, much NLP work has been devoted to detecting errors in the writing of non-native speakers learning English as a Second Language (ESL). These efforts have focused primarily on the main errors that ESL writers typically make, such as determiner usage, e.g. "We read *a* same book" (Han et al., 2006; Lee and Seneff, 2006; Nagata et al., 2006), preposition usage, e.g. "She is married *with* John" (Felice and Pullman, 2007; Gamon et al., 2008; Tetreault and Chodorow, 2008), and collocations, e.g. "We purchased a *strong* computer." (Sun et al., 2007).

While early grammatical error detection systems used a collection of manually-constructed rules (such as (Eeg-Olofsson and Knuttson, 2003)), recent ones are largely statistically-based. They work by first developing a model of correct usage based on well-formed text produced by native writers (usually news text). Next, the system flags a usage as an error if it has a low probability given the model. In essence, the system diagnoses as an error any usage that seems statistically unlikely given the probability of the correct usage. Optimally, statistical models should be trained on examples of incorrect usage as well as on examples of correct usage. However, the few annotated corpora of learner writing that do exist are either not freely available or are very small in size and thus insufficient for training large models.

There are, of course, problems that arise from training exclusively on error-free, native text. First, some errors are more probable than others. For example, in the ESL literature it is noted that many English learners incorrectly use "married with John" instead of "married to John". These observations are commonly held in the ESL teaching and research communities, but are not captured by current NLP implementations. Second, it is well known that ESL learners from different first languages (L1s) make different types of errors (Swan and Smith, 2001). For instance, a writer whose L1 is Spanish is more likely to produce the phrase "*in* Monday" while a German speaker is more likely to write "*at* Monday". Without errors in the training data, statistical models cannot be sensitive to such regularities in L1 error patterns.

In the absence of a large corpus of annotated non-native writing, we propose a novel approach which uses the "region" search found in both the Google and the Yahoo search APIs to compare the distribution of a certain English construction in text found on web pages in an English-speaking country to the distribution of the same English construction on web pages in a predominantly non-English speaking country. If the distributions differ markedly, this is a sign that the English construction may be problematic for speakers of that L1.

Consider the example in Table 1 of "depends on" and "depends of". Native writers typically use the preposition *on* in "depends on". It should be

Region	*on*	*of*	Ratio	RR
US	92,000,000	267,000	345:1	
France	1,500,000	22,700	66:1	5.22:1

Table 1: Region Counts Example for "depends *preposition*"

noted that one can construct examples with *of* such as "it depends *of* course on other factors..." though these happen much less frequently. This distribution is reflected in the region counts for the United States. The more common usage "depends on" is used 345 times more frequently than "depends of." However, when performing the same queries with France as the region, the ratios are considerably different: 66 to 1. This means that the ratio of ratios (RR) comparing the US to France is about 5.2 to 1. We hypothesize that if speakers of a particular L1 had no problem with the construction, then the distribution would look similar to that of the US, but that a large RR, such as the one obtained for "depends of" signals a potential error. If enough L1s have distributions that deviate from the native English distribution, then that provides additional evidence that the construction may be problematic for non-native speakers in general.

Knowing what constructions are problematic can allow us to tune a system trained on native text in different ways. One approach is to adjust internal thresholds to make the system more sensitive to known errors. Another is to augment the training data for the statistical model with more examples of correct usage of the construction.

This paper makes the following contributions:

- A novel approach to detecting common errors by non-native speakers of English that uses the "region search" in search engine APIs. To our knowledge, this is the first NLP approach to use the region-dependent search. (Section 2)
- A preliminary validation study of the approach (Section 3)
- An empirical analysis of the issues involved when using region counts (Section 4)

Although this is a general method for discovering errors, here we will discuss its use with respect to preposition error detection in which the context licenses a preposition, but the writer used the incorrect one.

2 Region-Counts Approach

More formally, the approach works in the following manner. Given a construction (such as "married *preposition*" or "they used *determiner* stone"), do:

1. Select a gold standard region to compare against (either the US or the UK).
2. Select a set of non-native regions to query.
3. For each region, query the construction in its variant forms (e.g., "married to", "married of", "married with"; "they used stone", "they used a stone" and "they used the stone.") using a search engine and save the counts.
4. Upon completion of step 3, find the most frequently occurring variant in the gold standard distribution and calculate the ratio of that variant compared to every other variant in the region.
5. Using the variant form that was most frequent in the gold standard distribution (e.g., "married to"), calculate for every other region the ratio of that variant's frequencies compared to each of the other variants' frequencies.
6. Calculate the RR by comparing the ratios in the non-native region to the corresponding ratios in the gold-standard region.
7. Use a threshold function on the RRs to flag a construction as problematic in a specific region or problematic in general. For details on setting the threshold function see Section 5.

To illustrate how the approach works, we will use the example construction "married *preposition*" using the Yahoo search engine API, three prepositions (*to*, *for*, *with*), the UK as the gold standard region, and three non-native regions (China, Russia, France). Table 2 shows the results of the approach with this construction's three variants. The columns labeled "Count" show the Yahoo web counts for that region and variant. In this example construction, *to* is the most frequent variant in the gold standard region, so for each region, the ratios are calculated: *to*:*for* and *to*:*with*. The figures are shown in the columns labeled "Ratios." Next, the RR is calculated between the non-native ratios and the gold-standard ratios. For example,

	to	*for*			*with*		
Region	**Count**	**Count**	**Ratio**	**RR**	**Count**	**Ratio**	**RR**
UK	6,200,000	1,050,000	5.90:1		1,890,000	3.28:1	
China	417,000	62,300	6.69:1	0.88:1	92,900	4.49:1	0.73:1
Russia	378,000	57,100	6.62:1	0.89:1	185,000	2.04:1	**1.61:1**
France	191,000	23,600	8.09:1	0.73:1	162,000	1.18:1	**2.78:1**

Table 2: Example of Approach on “married *preposition*” where *to* is the most frequent gold standard preposition

RR for “married for” (China) is 5.90:1 to 6.69:1, or 0.88:1.

A RR greater than 1 signals that the region uses that particular variant relatively more than the gold-standard region. The larger the RR, the greater the “over” usage of that form. For example, France’s ratio of “married with” versus “married to” is 2.78 times that of the UK. This is not surprising since many speakers of Romance languages have difficulties with the preposition *of*. Determining a threshold function for the RR (or any other metric one can derive from the relative frequencies) is an area we are currently exploring. One approach is to flag an entire construction if several regions have RRs markedly over 1.00, or if one variant has values over 1.00 in several regions. An example of this is “married with” which has a RR greater than 1.00 in two of the three regions in Table 2.

To put this approach into practice, one first needs to generate a list of constructions (and then variants), and use the region counting approach above to iterate through the list. In the case of preposition error discovery, one could take a large corpus of student writing and extract all bigrams (or any n-grams or skip-grams) that start with a preposition or end with a preposition, and treat those as constructions.

3 Proof of Concept

3.1 Validation with Examples of Known Errors

To test how well the approach described in Section 2 fares, we conducted a simple pilot study in which we checked to see if it was able to “discover” common errors described in the ESL literature. We collected 20 examples of common preposition errors from ESL research websites and second language acquisition papers. The examples consisted of the error commonly made, as well as the correct form. For the sake of space, we will focus on 5 of the 20 examples (see Table 3). The results for these 5 were representative of the larger set.

Correct Usage	Incorrect Usage
depends on	depends of
surprised by	surprised with
married to	married with
arrive at	arrive to
worried about	worried with

Table 3: Typical ESL Error Constructions

For each example, we collected region counts via Yahoo for 12 non-native regions, as well as counts for the US, which served as the gold-standard region. In all 20 examples, at least one region had a RR greater than 1.00. In 10 of the examples, over half of the regions had RRs greater than 1.00. Finally, in 15 of the 20 examples, at least one region had an RR greater than 2.00.

3.2 Validation with Student Data

Next, we checked to see if these errors actually occur in a large corpus of student writing and then quantified the need for error data in a preposition error detection system.

We extracted sentences which contained the target construction variants from 530,000 essays written for the Test of English as a Foreign Language (TOEFL®). The essays were written by non-native speakers representing 40 different L1s. Next, a trained annotator rated each construction variant, judging it as correct usage or incorrect usage, and then these judgments were reviewed by another trained annotator. Table 4 shows the corpus analysis and annotation statistics; for each construction the correct variant is listed first, and the incorrect variant second. The Frequency column shows the count for the variant in the entire corpus, and the Errors column gives the percentage of those cases that were judged to be an error by the annotator. For constructions with hundreds of cases, the annotator rated a randomly selected sample of 150.

Variant	Frequency	Errors
depends on	18,675	0.6%
depends of	813	97.3%
surprised by	221	3.3%
surprised with	61	34.4%
married to	82	9.8%
married with	134	93.3%
arrive at	1,201	12.6%
arrive to	871	95.3%
worried about	2,857	2.7%
worried with	36	91.7%

Table 4: TOEFL Corpus Analysis

All 20 constructions appeared in the corpus of student essays. More importantly, the corpus analysis validates what the ESL literature (and the region-counts approach) predicted: in four out of the five cases listed above, the "incorrect" variant was an actual error over 90% of the time.

3.3 System Performance

Next, we used a preposition error detection system to determine how many of these errors the system currently detects. If it correctly identifies most of the incorrect cases as errors, there is no need to augment the system with this procedure. On the other hand, if a system performs poorly on these errors, this then shows the extent to which the approach can potentially improve performance.

For this analysis, we used our preposition error detection system (Tetreault and Chodorow, 2008) trained on 7 million preposition examples from native text. The system has been shown to be among the best performing systems. Over all of the constructions, the system missed on average about 80% of the errors. Table 5 ("Original Model") shows the results for five of the constructions. While the system had very high precision, its recall was very poor. For example, for the "married with" variant, it missed 88% of the errors in the annotated corpus. We believe that this shows the potential benefit of increasing the sensitivity of the system to errors which are known to occur frequently in ESL writing.

One method of using the approach to improve a system is to build small models specifically tuned to handle those constructions. If the variant is encountered, the system uses the tuned model, otherwise, it uses the more general, original model. For each construction, we extracted 50k examples from native text and trained a model in the same manner as the original model. We then evaluated this model on the error variants ("Tuned Model" in Table 5). Recall improved for four out of five cases, and substantially for "depends of" (45.2% to 80.1%) and "married with" (12.4% to 48.7%). This is, of course, a very simple way of leveraging the region-counts approach into a system; there are more sophisticated machine learning approaches one could use to tune a smaller model or augment the original model, though this is outside the scope of the current paper. However, we believe that the gains from this straightforward model tuning show the potential benefit of increasing the sensitivity of the system to constructions in which errors are known to occur frequently in ESL writing.

4 Reliability of Web Counts

While web counts have the advantage of being free, Kilgariff (2007) observed that there are limitations associated with their use: (1) there is no lemmatizing or part-of-speech tagging, (2) search syntax is limited, (3) the number of queries per day is constrained by the search engine and (4) web counts are for pages, not for unique instances (a page could have more than one instance of the query term). Despite these problems, previous work (such as (Keller and Lapata, 2003; Lapata and Keller, 2005; Nakov and Hearst, 2005; Nakov, 2007)) has shown that different NLP applications can be improved by using web counts. In this section, we examine the extent to which the limitations commonly associated with general web counts also affect region web counts and thus our approach. In 4.1, we examine how variable the region counts are over the course of one week, and in section 4.2 we look at a sample of web pages that the region search method returns and assess the quality of the sample with respect to our approach.

4.1 Variability of Web Counts

Web counts tend to vary from week to week, and sometimes even from hour to hour. This can be a problem for any approach, such as ours, which

			Original Model		Tuned Model	
Variant	**Frequency**	**# of Errors**	**Precision**	**Recall**	**Precision**	**Recall**
arrive to	149	142	100.0%	20.4%	100.0%	35.2%
depends of	150	146	100.0%	45.2%	100.0%	80.1%
married with	122	113	100.0%	12.4%	99.1%	48.7%
surprised with	61	21	85.7%	27.3%	100.0%	27.3%
worried with	36	33	100.0%	57.0%	100.0%	60.0%

Table 5: System Performance on Error Constructions

assumes that the counts are fairly stable. A frequency spike or dip in one region count could skew a RR and thus an error may be missed or spuriously flagged.

To assess the variability of the counts, we took the 20 examples from the previous section and collected the respective region counts (with UK as a gold standard and 12 non-native regions) using both Yahoo and Google. The process was repeated for seven consecutive days allowing us to track the variability of 520 region counts[1]. For each region and variant combination, we calculated its coefficient of variation (CV) over the 7 days (i.e., σ/μ, the result of dividing the standard deviation of its counts by its mean count) and then averaged all 520 coefficients of variation. Yahoo and Google had average CVs of 0.02 and 0.08, respectively, suggesting that the Yahoo search engine's region counts were somewhat more consistent over that one week period.

The most variable Yahoo searches were "insisted on" (Sweden) with a CV of 0.23, "disgusted with" (China), with a CV of 0.21, and "confronted with" (France), with a CV of 0.20. Google's most variable searches were "love with" (Japan) with a CV of 0.92, "confronted with" (Poland) with a CV of 0.74, and "surprised by" (Russia) with a CV of 0.72.

Taken as an aggregate, the CVs look acceptable, however there were several individual queries that showed wide variation when repeated. In the Google experiments, 10% of all the queries had an average CV greater than 0.20. These results suggest that our approach will likely miss some potential errors (or produce false positives on others). One way of dealing with this is to repeat the experiment several times over the course of a week or month and select the constructions which are consistently flagged as an error across those days. Of course, while this approach has the advantage of flagging errors more reliably, it has the drawback of having to use one's daily search quota on repeating experiments, thus slowing the pace of discovering new errors.

4.2 Web Page Quality

While the variability of web counts can be an issue, the quality of the web pages counted in those hits can also impact the usefulness of the approach. For instance, it is possible that a variant with a high RR may not really be used incorrectly and that the high RR may be caused by missed punctuation, spam sites which repeat English phrases over and over, or American or British websites being hosted in a non-native region.

To determine the quality of the web counts, we randomly selected 10 variants with a very high RR and then examined the top 50 web pages that contained the variant, and another randomly selected 50, for a total of 100 web pages per variant. We annotated each web page using the scheme shown in Table 6. The third column of Table 7 lists the RR as well as the web counts for that variant.

The final tag distributions for each variant are shown in Table 7. Several of the variants: "confronted to", "consist by", "depend from", and "key-of", showed very high error counts (all 25% or more) which shows that for these cases the Ratio of Ratios metric is finding preposition usage examples that are problematic for non-native speakers. However, there are several other variants that were ranked highly that had very few errors. For example, "arrive on" had only four incorrect usages, and the remainder were either acceptable or language issues. Interestingly, many of the web pages in the set were tourism websites dealing with traveling to France. Another French example that only had a few errors was "nice on". We found that the overwhelming ma-

[1] There are 20 examples of 26 queries each: each example has a correct and incorrect construction, and 13 regions are queried for each.

Tag Name	Code	Description
Error	Err	The variant in the gloss is an example of an incorrect preposition usage
Acceptable	Acc	The variant in the gloss is an example of correct preposition usage
Garbage	Gar	Web page is a spam site or listed as an attack site by Firefox
Language	Lang	Variant is actually an acceptable string in the native language and is included in the count though page is composed of mostly English sentences
Repeated	Rep	Gloss appears in another website
English	Eng	Site appears to be an American or British site hosted in that region
Punctuation	Punct	The variant in the gloss has punctuation in the middle that was skipped over by the search engine, or there should have been punctuation between the two words.

Table 6: Web Quality Annotation Scheme

Variant	Region	RR	Count	Err	Acc	Gar	Lang	Rep	Eng	Punct
arrive on	France	5.65	629,000	4	75	1	16	3	1	1
confront of	China	7.55	186	15	17	34	0	23	0	2
confront to	Japan	15.64	1,470	15	22	30	0	14	0	8
confronted to	France	20.41	32,800	98	1	0	0	1	0	1
consist by	China	23.55	1,660	32	2	50	0	15	1	2
depend from	Russia	4.35	3,630	81	4	7	0	5	0	1
dreamt for	France	17.15	12,400	9	12	1	0	78	0	0
dreamt in	Poland	39.76	4,290	9	22	0	0	68	0	1
key of	Korea	6.26	507,000	25	61	0	0	11	0	1
nice on	France	8.81	199,000	5	84	6	3	3	0	7

Table 7: Quality of Sample Web Pages

jority of acceptable cases were actually about the French city *Nice* and not the adjective. Other variants showed other peculiarities, and thus highlights the danger of using the raw web counts blindly. The variant “dreamt for” received a high “repeated” count because it is the title of a music album (“Dreamt for Light Years in the Belly of a Mountain”), and many French websites that were counted were either selling or reviewing the album. A similar trend happened when searching the string in the US or UK regions, but the ratio was larger for the French site because the counts for the other “dreamt *preposition*” variants were relatively smaller.

Overall, this quality experiment showed that for all ten cases, there were indeed some errors in each of the 100 glosses. However, some of the cases were very weak and were affected by problems with repeated website, punctuation and language issues.

Next, we checked how often each of the ten constructions appeared in our corpus of 530,000 student essays and, as in Section 3, rated each case as correct or incorrect preposition usage. Table 8 shows the frequency and error rates. The rightmost column shows the percentage of glosses that had the construction as an error (from Table 7, column 5). The chart shows that in 8 of the 10 constructions, a majority of the cases were actually errors. And in the remaining two, at least 20% of the cases were errors. It is also notable for those two cases that the web error counts were quite low: 4.0% and 5.0% respectively. This probably means that L1s other than French also use those phrases incorrectly.

5 Related Work

The “region counts” approach is just one method of trying to enhance current error detection models. For instance, Foster and Andersen (2009), created a system (GenERRate) to insert errors into native corpora to create large amounts of artificial non-native-like corpora. The advantage of their system is that it allows the user to create style sheets that control the type and number of errors. However, the performance impact from using artificial corpora in the error domain has yet to be ex-

Construction	Freq.	% TOEFL Errors	% Web Errors
arrive on	70	27.2%	4.0%
confronted to	100	100.0%	15.0%
confront of	11	72.8%	15.0%
confront to	21	90.5%	98.0%
consist by	8	100.0%	32.0%
depend from	94	91.5%	81.0%
dreamt for	8	75.0%	9.0%
dreamt in	3	100.0%	9.0%
key of	96	96.0%	25.0%
nice on	22	22.3%	5.0%

Table 8: Corpus Analysis of Discovered Errors

amined closely. Hermet and Désilets (2009) also developed a novel method of using roundtrip Machine Translation techniques to improve a standard preposition error detection system. Although their evaluation corpus was limited to 133 prepositions, the hybrid system outperformed their standard method by roughly 13%.

6 Discussion

In this paper, we have presented an approach to detecting common grammatical errors found in the writing of ESL speakers. The approach involves using the region search function found in the Yahoo and Google search APIs to gather statistics on the distribution of potentially problematic constructions in different non-native regions. These distributions are then compared to the distribution of a native English region. In addition, we presented results from a pilot study that showed the approach can detect common ESL errors noted in the literature, and we also verified that these errors do in fact appear in a large corpus of varied student writing, but that a state of the art preposition detection system fails to detect a significant portion of these errors. Finally, we demonstrated that these systems can easily be improved by training models that target the specific constructions. We believe that this demonstrates the potential impact such an approach can have on a system which detects common ESL errors.

While the preliminary results appear encouraging, our analysis showed that problems with variation as well as the quality of English web pages counted in non-native region searches may reduce the effectiveness of the approach. As a result, our future work will focus on the following areas:

Similarity Function In this work, we have used the RR metric to compare one region's variant ratios to the gold standard's, but other measures of distributional similarity are also available, such as Cosine similarity and Kullback Liebler (KL) Divergence.

Thresholding Function Another area to explore is how to threshold the similarity function. One could flag a whole construction or variant if several regions have RRs over a set value. This function can be empirically determined by comparing the distributions of constructions known to have errors with those that are known to be non-problematic for non-native speakers.

Collapsing Regions The variability in the region counts has the effect of potentially skewing the results of the thresholding function. False positives can arise if one uses a threshold function that flags the whole construction as an error if only a few regions have a very high RR. One way of reducing the impact of variable regions is to collapse regions into different language groups: East Asian (Japan, Korea, China), Slavic (Russia, Poland), and Romance (France, Spain, Italy, etc.). One can carry this aggregation even further and group all non-native regions into one class. The advantage of this approach is that it is less sensitive to the usual variations from the search engines, but the effects due to smaller regions may be less detectable and thus the system will miss these cases.

Finally, it should be noted that although we have focused on preposition error detection in this paper, this is a general approach that can discover problematic constructions for other types of errors. The method also has applications beyond grammatical error detection. For instance, it can form the foundation of a system which automatically generates test items for ESL students.

Acknowledgments

We would like to first thank Tomonori Nagano for his technical assistance and Sarah Ohls for her annotation work. In addition, we thank Michael Gamon, Detmar Meurers and the three anonymous reviewers for their comments and feedback.

References

J. Eeg-Olofsson and O. Knuttson. 2003. Automatic grammar checking for second language learners - the use of prepositions. In *Nodalida*.

R. De Felice and S. Pullman. 2007. Automatically acquiring models of preposition use. In *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*.

J. Foster and Øistein Andersen. 2009. Generrate: Generating errors for use in grammatical error detection. In *The 4th Workshop on Building Educational Applications Using NLP*.

M. Gamon, J. Gao, C. Brockett, A. Klementiev, W. B. Dolan, D. Belenko, and L. Vanderwende. 2008. Using contextual speller techniques and language modeling for ESL error correction. In *IJCNLP*.

N-R. Han, M. Chodorow, and C. Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12:115–129.

M. Hermet and A. Désilets. 2009. Using first and second language models to correct preposition errors in second language authoring. In *The 4th Workshop on Building Educational Applications Using NLP*.

F. Keller and M. Lapata. 2003. Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29(3).

A. Kilgarriff. 2007. Googleology is bad science. *Computational Linguistics*, 33(1).

M. Lapata and F. Keller. 2005. Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing (TSLP)*, 2(1):1–31.

J. Lee and S. Seneff. 2006. Automatic grammar correction for second-language learners. In *Interspeech*.

R. Nagata, A. Kawai, K. Morihiro, and N. Isu. 2006. A feedback-augmented method for detecting errors in the writing of learners of English. In *Proceedings of the ACL/COLING*.

P. Nakov and M. Hearst. 2005. Using the web as an implicit training set: application to structural ambiguity resolution. In *HLT-EMNLP*.

P. Nakov. 2007. *Using the Web as an Implicit Training Set: Application to Noun Compound Syntax and Semantics*. Ph.D. thesis, EECS Department, University of California, Berkeley.

G. Sun, X. Liu, G. Cong, M. Zhou, Z. Xiong, J. Lee, and C.-Y. Lin. 2007. Detecting erroneous sentences using automatically mined sequential patterns. In *ACL*.

M. Swan and B. Smith, editors. 2001. *Learner English: A teacher's guide to interference and other problems*. Cambridge University Press.

J. Tetreault and M. Chodorow. 2008. The ups and downs of preposition error detection in ESL writing. In *COLING*.

Extracting domain terminologies from the World Wide Web

M. Wendt, C. Büscher, C. Herta, M. Gerlach,
M. Messner, S. Kemmerer, W. Tietze and **H. Düwiger**
neofonie GmbH
Berlin, Germany
{wendt, buescher, herta, gerlach, messner,
kemmerer, tietze, duewiger}@neofonie.de

Abstract

Domain specific terminologies are an important starting point for the automatic extraction of ontologies. In this paper, we present an industrial strength application for creating such terminologies from the World Wide Web. Using raw Web data for terminology extraction poses the challenge of dealing with noise of various types. We show how de-duplication and topic-filtering methods can be used to build clean domain and reference corpora. We combine statistical methods to extract German single- and multi-word terms. Also, we introduce an extension of Broder's de-duplication method for fast online filtering.

1 Introduction

Terminology extraction is the prerequisite for all aspects of ontology learning from text (Buitelaar et al., 2005). While domain ontologies are valuable resources for building (vertical) semantic search applications, domain specific terminologies (e.g. biology, medicine, tourism etc.) can also be used directly for various purposes like tagging of documents with keywords, annotation of documents with searchable meta-information for improving retrieval quality or routing in meta-search applications.

In this paper we present a methodology for extracting domain terminologies from the Web. The work conducted on this topic is part of a larger research project on industrial-strength ontology population. As a proof of concept, the medical and health care domain was chosen. The resulting ontology will be integrated in a patient information system. Therefore, the methodology presented here will be applied to the medical and health care domain. Nevertheless, our primary aim was to develop a process applicable for arbitrary domains.

For terminology extraction, we present our method for building a domain corpus by domain-focused crawling and topic filtering. After identifying candidate terms by chunk-parsing, the domain terminology will be extracted by comparing the term statistics of the domain corpus to the statistics of a reference corpus. An important aspect in this context is the treatment of multi-word terms. Although candidate phrases are reliably identified by the chunk parser, not every phrase constitutes a fixed domain term, even if it appears significantly more often in the domain corpus. Therefore, it is important to distinguish between phrases which are generated productively (e.g. "good doctor") and fixed terminological expressions ("diabetes mellitus"). To achieve this, we examined the applicability of a phraseness measure derived from a bigram language model.

The advantage of directly using Web documents, instead of human-curated corpora, is the availability of a great variety of information, covering several different domains and languages. However, this poses several problems as the Web is by no means a clean source of textual data. First, we face the problem of noisy input resulting from navigation elements, headers, footers, advertisements, as well as forums, login and error pages or pages that are merely generated for search engine optimization. Second, the Web contains many duplicate pages. We will show how content extraction and duplicate-filtering techniques can be used to successfully tackle these problems.

2 Related Work

The Web has been used as a source of linguistic data for various applications in natural language processing for quiet some time now (Kilgarriff and Grefenstette, 2003). Recent technical advances have made it feasible for researchers to create large general language and specialized corpora by combining Web crawling, text filtering and linguistic post-processing (Baroni and Kilgarriff, 2006), (Baroni and Ueyama, 2006).

Most terminology extraction methods are statistical corpus based approaches (Pantel and Lin, 2001). However, the majority assume the existence of a clean input corpus (e.g. administered data from document warehouses) and thus, avoid many problems that arise when crawling the Web to build the corpus. For example, Wermter and Hahn (2005) extract terminology from a large biomedical text corpus and explore the distinction of specific terminology from common non-specific noun phrases. Navigli and Velardi (2004) use terminology extraction methods for ontology population, using input data from dedicated Web sites and data warehouses. Velardi et al. (2008) also use terminology extraction as a first phase in building domain specific glossaries.

Most of the methods for handling multi-word expressions are based on language models using n-grams (Manning and Schütze, 1999). Dunning (1993) shows how statistical tests on different distribution assumptions can be used to filter collocations. Tomokiyo and Hurst (2003) use information theoretic measures and present a combined solution to measure both strength of collocation (*phraseness*) and domain-specificity (*informativeness*). Frantzi et al. (1998) combine linguistic and statistic information in a domain-independent method for extracting multi-word terms. Other kinds of linguistic pre-processing can be used to narrow the search space for collocations down to noun phrases (Justeson and Katz, 1995).

3 System Overview

Figure 1 shows the three main phases in the terminology extraction process. In the first phase, the domain- and the reference corpus are built by initial Web crawling and cleaning the documents from HTML markup and irrelevant content. After detecting near duplicates and filtering out non-domain documents using a text classifier, the target and reference corpus are both stored for later processing.

The second phase uses documents from both corpora as input data and detects candidate terms by extracting noun phrases from the domain corpus. Additionally, the relevant term frequency statistics are collected. By using a discriminant function on metrics based on the term statistics of both corpora, we eliminate term candidates that are not domain specific.

In order to remove the remaining bogus terms from the terminology, in a third phase, we apply additional filter heuristics to eliminate them. Filtering also involves removing irrelevant multi-word phrases.

4 Building the Corpora

To extract a domain terminology from a *domain corpus*, a second corpus containing non-domain documents is necessary. This is the *reference corpus* and the frequency statistics of terms in both corpora are used to distinguish between relevant and non-relevant domain terms. The corpus-building approach taken here is composed of four steps: crawling, extraction of the document contents, de-duplication and classification (not for the reference corpus).

4.1 Crawling

We used an open source Webcrawler[1] for crawling the Web. Crawling for the domain corpus requires a focus on domain-relevant documents. We restricted the crawling process to 200 hand-selected German Web sites from the medical and health sector, comprised of large Web portals with a high ratio of user generated content. For the reference corpus, we selected several large German Web portals as initial seeds and conducted a breadth-first crawl to collect a diverse selection of documents from the German top-level domain `*.de`. For our experiments, we crawled approx. 6 Mio. documents from medical domains and approx. 8 Mio. documents from non-medical domains.

4.2 Content Extraction

In order to extract plain text from the HTML-pages found on the Web, we used an open source HTML-parser[2].

However, stripping the documents off HTML mark-up alone does not suffice, as the documents do not only consist of proper body text, but also include much additional information.

To handle this, our text extraction component uses heuristics to identify the content-bearing passages of a page. It processes the paragraph structure of the documents which is identified during HTML-parsing. Each contiguous partition of the

[1] Heritrix, `http://crawler.archive.org`

[2] Jericho HTML Parser, `http://jerichohtml.sourceforge.net/doc/index.html`

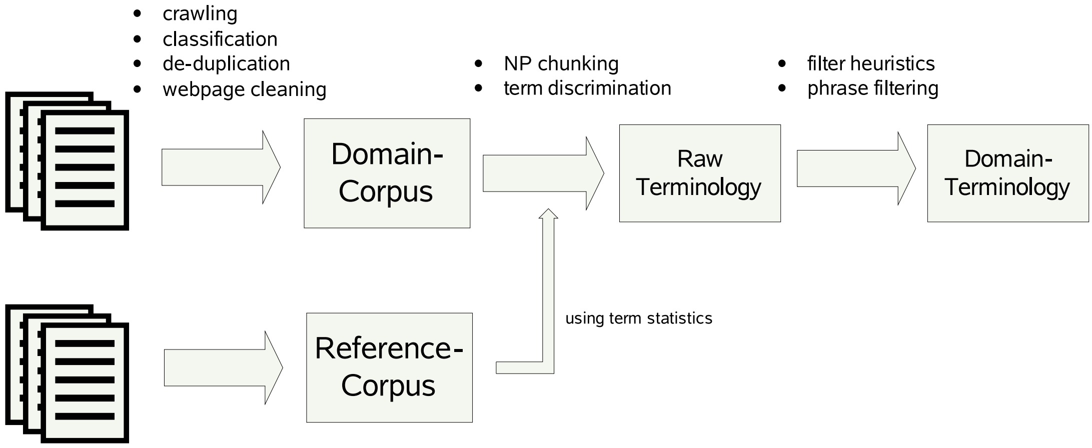


Figure 1: System Overview

plain text, including ample text sections, as well as headers and lists items, is defined as a paragraph. We retain a paragraph for subsequent processing if:

1. it contains at least s sentences
2. or if all of the following are true:
   - it contains at most c percent of capitalized words
   - it contains at least t_l tokens
   - it contains at most t_u tokens
   - it ends with punctuation

For our experiments we experimentally found that the following parameters yielded the best results: $s = 2$, $c = 70$, $t_l = 5$, $t_u = 20$.

4.3 Near Duplicate Detection

Limiting the domain crawl to hand selected Web sites leads to a higher yield of domain documents. However, one remaining disadvantage of using Web data is the high number of identical or nearly identical documents. For this reason, we implemented the near duplicate filtering method presented by Broder (2000) and modified this method for fast online de-duplication. Almost 50% of our documents were discarded as near-duplicates in this phase. Near-duplicates consisted mostly of automatically generated pages like login, error and navigation pages.

Modifications of Broder's Method for Rapid Online De-Duplication

Broder's method for de-duplication is a two-step computational process: (1) computing the fingerprint of each document and (2) finding out which documents belong together in a bulk computation. However, this approach assumes that there is a fixed corpus of documents being built all-at-once. In contrast, we were interested in a way to filter out duplicates online, that is, directly in the corpus building pipeline. This way we could keep the corpus duplicate-clean while retaining the flexibility needed when the corpus has to be held up-to-date with the Web.[3]

We implemented the fingerprinting algorithm used in Broder (2000) that generates a fixed-size fingerprint fp of n *super shingles* $s_i, i = 1 \ldots n$ (also called the *super sketch*) for each document:

$$fp = \{s_1, \ldots, s_n\}$$

The fingerprint can be conceived as a condensed representation of the document's content. The relation of being near-duplicates $sim(d_1, d_2)$, henceforth called *similarity*, between two documents d_1 and d_2 holds, if more than a given threshold θ of super shingles at the same index match.

For detection, Broder (2000) suggests storing the super shingles of each document as pairs $\langle s_i, d \rangle$ of the shingle value s_i[4] and document id d. In the detection phase the table is sorted by the shingle value, creating a list of matches in the format $\langle d_1, d_2 \rangle$ (the ids of two documents). In the worst case[5], the list of document pairs may be quadratic to the number of documents; therefore, the super sketch must be kept small. The matches

[3] This requirement also arises in other contexts of industrial scale information retrieval.

[4] Keep in mind that only shingles at the same index must be compared.

[5] This case only occurs, if all of the documents are duplicates.

are then merge-sorted by document ids and only those matches having a count greater than θ are kept.

Afterwards, implicitly presuming the property of transitivity for the similarity relation[6], Broder (2000) suggests computing the union of each set of duplicates. Note, that as a consequence, if $sim(d_1, d_2)$ and $sim(d_2, d_3)$, then $sim^+(d_1, d_3)$ will hold (as a result of transitive closure), even if d_1 and d_3 are not similar according to their fingerprints.

In order to adapt the aforementioned approach to online near-duplicate filtering, we created a database containing triplets $\langle i, s_i, d\rangle$ of the shingle index i, shingle value s_i and document id d, making use of an index to retrieve the matches. In the corpus-building pipeline, each document is fingerprinted and the fingerprint is then stored in the database. Similar documents are retrieved from the database making use of the index.

Just as the match list in the original approach grows quadratic in the worst-case, by applying a straightforward implementation, the number of duplicates retrieved for each document grows[7], successively slowing down the pipeline. It is interesting to note that the duplicates found for each set where almost invariably the same. To overcome this issue, we divided the fingerprint database into two tables: one for storage and one for lookup. For each duplicate set, a representative document was chosen virtually at random, otherwise the lookup table was kept clean from duplicates. It should be obvious that this modification avoids the worst-case complexity of $O(n^2 f(n))$ by lowering the complexity to $O(nf(n))$, where n is the number of documents and $f(n)$ is the complexity of the index lookup.

It is important to understand, that while this modification also presumes the transitivity property of sim mentioned above, it is in fact inverting its consequences. Suppose $sim(d_1, d_2)$ has been detected first and $sim(d_2, d_3)$ also holds, then $sim(d_2, d_3)$ will only be detected, if d_2 is the representative of the set $\{d_1, d_2\}$. However, keeping in mind that the original duplicate filtering method is approximative in nature, we found this modification to work satisfyingly in practice.

4.4 Web Site Classification

Narrowing the crawl to the target domain (see 4.1) still does not reliably assure that all documents crawled belong to the target domain. To classify the documents, we used a Support Vector Machine (SVM) (Chang and Lin, 2001), an algorithm that is considered state of the art for this type of task (Joachims, 2002).

We trained a binary SVM classifier on 4,000 manually labeled domain and non-domain documents randomly sampled from the crawl using a RBF-Kernel. Our training features consisted of a list of approx. 500 German hand-selected, frequently occurring words from the medical domain (like “Arzt” (doctor), “Patient” (patient), “Blinddarm” (appendix), “Schlaganfall” (stroke)) and a list of 65 regular expressions matching frequent word prefixes and suffixes frequently found in medical terminology (like “*ologie”, “*skopie”, “*etis” or “*pathie”). We used the term frequency (TF) of the features in each document’s feature vector.

The performance of our classification model was evaluated using 200 manually-labeled test documents randomly drawn from our crawls. We reached an overall accuracy of 91% with 89% precision and 83% recall for the “medicine”-class, which corresponds to an F1-Measure of 86%.

5 Extracting the Domain Terminology

In this step, we use the corpora, aforementioned in 4, to extract relevant terms for the domain terminology. First we extract candidate phrases from domain documents, then we filter these candidates using term statistics from both corpora. Finally we use a phraseness measure to exclude irrelevant multi-word phrases and apply other heuristics to achieve a high quality result terminology.

5.1 Candidate Extraction

To narrow our target candidates (see Justeson and Katz (1995) for motivation), we focused on extracting noun phrases using a chunk parser from a commercial NLP software kit[8]. The module identifies noun phrase chunks on the basis of POS-tag patterns. The chunks selected by the parser come without an article and may also have a complex structure containing other chunks. Complex chunks arise in the case of close appositions

[6] Note that similarity is distinct from resemblance of which Broder explicitly states that it is not transitive.

[7] Duplicate sets found in practice had a size of up to 50,000 documents.

[8] Inxight LinguistX

like "Professor Curt Dihm"[9] and in some limited cases of prepositional phrases like "Entstehung von Neurodermitis" (*development of neurodermatitis*).

In the case of complex chunks, sub-chunks where also extracted. Regarding the term statistics used in 5.2, all sub-chunks where counted in addition to their parent. We used the lemmatized form of the candidate terms as an identifier to merge the different surface forms.

5.2 Domain-Term Filtering

After extracting the candidate terms, the next phase is concerned with recognizing the terms that are relevant for the domain corpus.

Following Velardi et al. (2001) we use the concepts of "Domain Relevance" and "Domain Consensus" which we will briefly introduce. Both scores are used in a linear decision function that discriminates between domain and non-domain terms.

Domain Relevance

The intuitive idea of Domain Relevance definition by Velardi et al. (2001) is to compare the frequency of a candidate term across different corpora. A term that is relevant for a target domain is expected to occur more often in the corresponding domain corpus D_{dom} than in a general reference corpus D_{ref}. Let T_D contain all term candidates t extracted from documents d in corpus D. The estimated conditional probability $P(t|D)$ of a candidate term $t \in T_D$ given corpus D is then:

$$P(t|D) = \frac{cf(t,D)}{\sum\limits_{t' \in T_D} cf(t',D)}$$

Here $cf(t,D)$ is the collection frequency (the total number of occurrences) of t in corpus D. The Domain Relevance DR of term t with respect to the domain corpus D_{dom} is then defined as:

$$DR_{t,D_{dom}} = \frac{P(t|D_{dom})}{P(t|D_{dom}) + P(t|D_{ref})}$$

Accordingly, the Domain Relevance is maximally 1 if the term candidate only appears in the domain corpus. On the other hand, if $DR < 0.5$, then the term frequency is higher in the reference corpus. A score between 0.5 and 1 is assigned to terms that appear in the domain corpus with a higher probability than in the reference corpus.

Domain Consensus

While the Domain Relevance score compares term frequencies across different domains, the Domain Consensus measures the distributed use of a candidate term in the domain corpus only. The intuition behind this metric is that the use of a term across many documents in the domain expresses a certain consensus about the importance of that term in the domain. In contrast, terms appearing in fewer documents are regarded to be less important for the domain. Velardi et al. (2001) consider the distribution of term t across all documents $d \in D_{dom}$ which they define as:

$$P_t(d) = \frac{tf(t,d)}{\sum\limits_{d' \in D_{dom}} tf(t,d')}$$

Here $tf(t,d)$ is the term frequency of t in document d. The Domain Consensus $DC_{t,D_{dom}}$ is then defined as the entropy H of this distribution:

$$\begin{aligned} DC_{t,D_{dom}} &= H(P_t(d)) \\ &= \sum_{d' \in D_{dom}} \left(P_t(d') \log(\frac{1}{P_t(d')}) \right) \end{aligned}$$

Decision Function

As a final decision score $f(t)$, we use an affine combination of Domain Relevance and Domain Consensus in which the factor α controls the contribution of both scores:

$$f(t) = \alpha \cdot DR_{t,D_{dom}} + (1-\alpha) DC_{t,D_{dom}} \quad (1)$$

We experimentally determined a threshold θ. Only terms with $f(t) > \theta$ were included in the final domain terminology. The tuning of α and θ was performed with respect to test data, and will be described in detail in section 6.1.

5.3 Phrase Filtering

Until now, we have discussed our method of determining the domain-specific terminology using methods of discriminant analysis. This method performs well for single-word terms, however, applying it to multi-word noun phrases identified by

[9] The pattern "Noun Noun" used here, is particularly error-prone in the case of German, often generating false positives. However, we tried to filter the noise generated by this pattern in the subsequent steps, especially when calculating the phraseness score later.

chunk parsing does not take into account the distinction between noun phrases generated productively (e.g. “guter Arzt”, engl.: *good doctor*) and fixed multi-word phrases (e.g. “Morbus Crohn”). We want to eliminate the former and keep the latter in the terminology. Also, we are interested in whether the word “Morbus” is a term in its own right.

To overcome this problem, we implemented the *phraseness measure* proposed by Tomokiyo and Hurst (2003), which our experiments showed to perform approximately the same as the log-likelihood ratio test (Dunning, 1993). The latter measure, however, does not permit the distinction of negatively correlated sequences from strongly correlated sequences if used straightforwardly. In addition, there is no straightforward application to sequences of arbitrary length.

Following Tomokiyo and Hurst (2003), we use a bigram language model[10] and measure its pointwise Kullback-Leibler divergence to the unigram model. The basic assumption behind this is that the bigram model better fits fixed terminological expressions, whereas, the unigram model will assign a higher probability to noun phrases that are generated productively.

Under the bigram model, a multi-word sequence $w = w_1 \dots w_n$ has a probability of $p(w) = \prod_{i=1}^{n} p(w_i|w_{i-1})$, while under the unigram model, it is $p'(w) = \prod_{i=1}^{n} p(w_i)$. The pointwise Kullback-Leibler divergence between both probabilities is defined as:

$$\delta_w(p||p') = p(w) \log \frac{p(w)}{p'(w)}$$

This measure was applied to all multi-word candidate terms during the phrase filtering stage in the terminology post-processing. The experimental results are discussed in 6.1.

5.4 Terminology Post-Processing

Comparing term frequencies according to the measures mentioned in 5.2 would probably suffice if the input corpus was noise-free. However, although most of the noise was already eliminated during content extraction (see 4.2) some errors remain. Nevertheless, implementing a more sophisticated content extraction would not be worthwhile, as some residual noise always remains.

term	**cf**
rein informativen Zwecken	189074
ausführlichen Nutzungsbedingungen	189066
Grundlage für Selbstdiagnosen	189066
Newsletter-Leser	109474
Onlinepharma48	92365
Apoversandpunkt	92365
Medikamente-per-Klick	92365
EU-Versandapotheke	92365
apondo.de	92365

Figure 2: Most frequent terms where $cf = n \cdot df$

The remaining noise in the raw terminology included terms typical for forums (e.g. “Beitrag” - *posting*), dates, user names (“tommy1983”), functional content (“RSS-Feed”, “Login”). To clean our terminology from such noise, we used first a customizable blacklist and some regular expressions for filtering tokens containing special characters or digits.

A second heuristic exploits the fact that most of the functional content regularly reoccurs on each page of a particular Web domain. As a result, the *collection frequency* cf of the unwanted terms was a multiple of the *document frequency* df. A probable explanation for this is that many Web pages in the domain corpus are generated by a CMS using templates, such that terms appearing in these templates have the same occurrence pattern in each generated document. In contrast, relevant terminology terms appear on different documents with varying frequency.

Figure 2 shows ten of the most frequent terms where $cf = n \cdot df$ for some $n \in \mathbb{N}$ from our data, which are all undesired. The phrases are used in imprints and disclaimers, often appearing across all pages of the same Web domain. The sample also contains names of online pharmacies that were regularly present in the advertisement segment of one particular Web domain.

Since none of these are interesting domain terms, we filtered out term candidates where $cf = n \cdot df$ in our experiments. Although this also filters out a few „good” candidate terms, it significantly reduced the noise in the result terminology.

6 Experimental Results

6.1 Term filtering

For evaluating our method for the identification of domain terms and tuning the parameters of the de-

[10] see (Chen and Goodman, 1996) for accurate n-gram modeling and smoothing

cision function (1), we viewed the task of term filtering as a binary classification problem. As test data we used approx. 6,500 terms, consisting of 1,200 single-token terms for the medical class, taken from the index of a medical lexicon. In addition, the 5,300 most frequent nouns in the German vocabulary[11] were used as negative examples.

We calculated specificity and recall with respect to the test data using the term frequency statistics from our domain and reference corpus and the decision function from section 5.2. Since the real data have an unknown skew, which is in general different from the skew in the test data, the metric "specificity" is preferred to that of "precision", since it is insensitive to skew.

The graph in figure 3 shows the specificity and recall for choosing different weights for α. The weight α for the domain relevance was varied for $\alpha \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.0\}$. For each setting, the threshold θ of the decision function was varied from 0 to 1. The graph shows that a changing weight on the Domain Consensus score in the decision function does not significantly influence the result quality for our data. To maximize both recall and specificity, setting $\alpha = 0.95$ and $\theta = 0.7$ minimized the Euclidian distance of the specificity/recall-value to their maximal value of 1 (represented by the upper right corner in the graph). These values were also used in the following experiments.

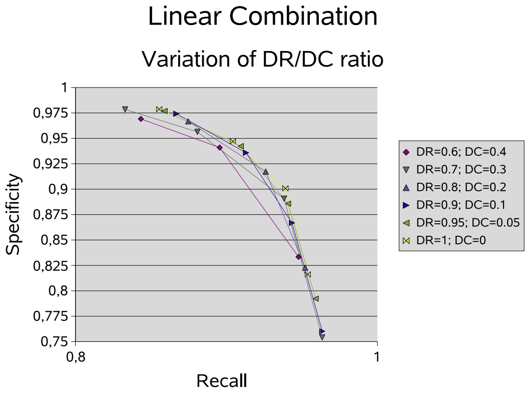


Figure 3: Evaluation of discriminant analysis parameter settings

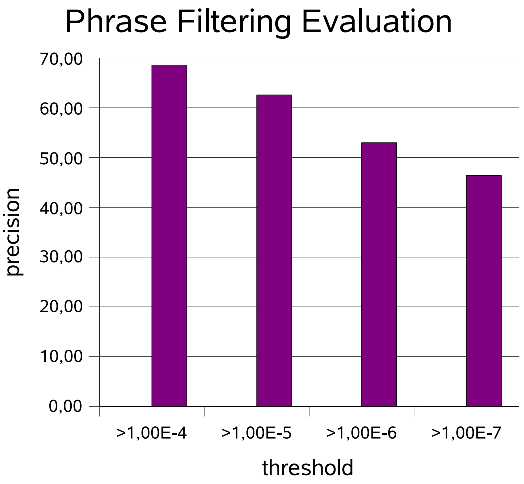


Figure 4: Phrase Filtering precision for various thresholds

Phrase Filtering Evaluation

To evaluate the phraseness measure described in 5.3 we first sampled 500 multi-word noun phrases from our terminology. The values were in the interval $[-10^{-4}, 10^{-3}]$. We manually divided the phrases into the two classes of productively generated vs. fixed phrases. We tried various thresholds to separate the two classes but were unable to find a value where the precision for the fixed phrases was significantly higher than 50%. Both classes seemed to be evenly distributed across large parts of the range of the phraseness score. However, we observed that, especially for large values ($> 10^{-4}$), there are more fixed phrases than phrases that are productively built. In a second experiment, we randomly sampled 50 phrases only above a threshold of $10^{-i}, i = 4 \ldots 7$, and repeated this experiment 10 times. Figure 4 shows the average precision of the fixed-phrase class. It can be observed that in the upper range the amount of phrases useful for the domain terminology is significantly higher than in lower ranges. Using a threshold of 10^{-5} in our experiments, we were able to extract 5,500 out of 61,000 multi-word phrases for our result terminology. Although this strict filtering reduces recall, it proved to significantly remove low-quality multi-word phrases from the terminology.

6.2 Result Terminology

After filtering the raw terminology obtained from the application of our discriminant analysis, the resulting terminology consisted of approx. 122,000

[11] Wortschatzsammlung Uni Leipzig `http://www.wortschatz.uni-leipzig.de/html/wliste.html`

Single-word terms	**Multi-word phrases**
Arzt (doctor)	änd Ärztenachrichtendienst Verlagsgesellschaft mbH (*a medical publisher*)
Patienten (patient)	gesunde Zähne (healthy teeth)
Fall (case)	praktische Tipps (practical hint)
Fragen (questions)	teilnehmende Ärzte (participating doctors)
Medikament (medication)	ambulante Chirurgie (emergency surgery)
Informationen (information)	vertragsärztlichen Versorgung (medical care by SIH-physicians)
Hinweis (advice)	neue Gebührenordnung (new physician fee schedule)
Diagnostik (diagnostics)	Heil- und Kostenplan (fee- and cost plan)
Empfehlungen (recommendations)	allergische Erkrankungen (allergic disease)
Therapie (therapy)	Thema Raucherentwöhnung (smoking withdrawal)

Figure 5: Top-10 single- and multi-token terms in result terminology

terms (about 50% of them being single-word terms) that occurred in 331,000 distinct surface forms in our corpus. Figure 5 shows the Top-10 terms with the highest collection frequency, single-word terms on the left and multi-word phrases on the right.

Using the 6.500 test terms from 6.1, we again evaluated the overall quality of the result terminology after post-processing, as described in section 5.4. Here we achieved an accuracy of 92% with 78% precision, 94% specificity and 81% recall.

7 Conclusion and Future Work

We have presented a method for extracting domain specific terminologies by crawling and processing Web documents. To yield a high-quality terminology directly from raw Web data, we combined different noise-reduction techniques. Near duplicate detection was implemented to prevent obtaining distorted term frequencies. To meet industrial-scale requirements, we modified the original algorithm for fast online de-duplication.

By applying a discriminant function based on term statistics of two corpora, we filtered domain relevant terms. We also examined the use of bigram statistics to filter out irrelevant multi-word phrases. We successfully applied the methodology for generating a German health terminology. To extract terminologies for different target domains, only the set of Web sites that are used as seeds for the crawler have to be changed. Also, a different classification model has to be trained. The extra work required for most domains will be minimal compared to the effort of creating domain specific terminologies manually.

As discussed in the introduction, the extraction of a domain terminology for the health domain was the first step in our research on methods for automatic ontology population. The quality of the results from our experiments encourages us to use the domain terminology as input data for our ongoing research. Additionally, we will extend our research to include topic specific crawling, and pursue the issues of Web site cleaning and identifying multi-word terminological expressions.

Acknowledgments

We would like to thank Scott Robinson for helpful comments and his revision of this paper. This project was supported by the German Federal Ministry of Economics and Technology on the basis of a resolution by the German Bundestag.

References

Marco Baroni and Adam Kilgarriff. 2006. Large linguistically-processed web corpora for multiple languages. In *Proceedings of EACL*. The Association for Computer Linguistics.

Marco Baroni and Motoko Ueyama. 2006. Building general- and special-purpose corpora by web crawling. In *Proceedings of the 13th NIJL International Symposium on Language Corpora: Their Compilation and Application*.

Andrei Z. Broder. 2000. Identifying and filtering near-duplicate documents. In *COM'00: Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching*.

Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. 2005. Ontology learning from text: An overview. In Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini, editors, *Ontology Learning from Text: Methods, Evaluation and Applications*, Frontiers in Artificial Intelligence and Applications, pages 3–28. IOS Press.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Stanley F. Chen and Joshua T. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual* *meeting on Association for Computational Linguistics*.

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19:61–74.

Katerina T. Frantzi, Sophia Ananiadou, and Jun-ichi Tsujii. 1998. The c-value/nc-value method of automatic recognition for multi-word terms. In *ECDL '98: Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, pages 585–604, London, UK. Springer-Verlag.

Thorsten Joachims. 2002. *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms*. Kluwer/Springer.

John S. Justeson and Slava M. Katz. 1995. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27.

Adam Kilgarriff and Gregory Grefenstette. 2003. Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29:333–347.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.

Roberto Navigli and Paola Velardi. 2004. Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics*, 30:2004.

Patrick Pantel and Dekang Lin. 2001. A statistical corpus-based term extractor. In *AI '01: Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence*, pages 36–46, London, UK. Springer-Verlag.

Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions*, pages 33–40, Morristown, NJ, USA. Association for Computational Linguistics.

Paola Velardi, Michele Missikoff, and Roberto Basili. 2001. Identification of relevant terms to support the construction of domain ontologies. In *ACL-EACL Workshop on Human Language Technologies*. Kluwer Academic Publisher.

Paola Velardi, Roberto Navigli, and Pierluigi D'Amadio. 2008. Mining the web to create specialized glossaries. *IEEE Intelligent Systems*, 23(5):18–25.

Joachim Wermter and Udo Hahn. 2005. Finding new terminology in very large corpora. In *K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture*, pages 137–144, New York, NY, USA. ACM.