NEALT

Northern European Association for
Language Technology

NEALT Proceedings Series No. 55

# Selected contributions from the Eighth Swedish Language Technology Conference (SLTC-2020)

25–27 November 2020, Gothenburg, Sweden and Online

Editors: Peter Ljunglöf, Simon Dobnik, and Richard Johansson

# The Eighth Swedish Language Technology Conference (SLTC-2020)

## Selected Contributions

25–27 November 2020
Gothenburg, Sweden
Online

Front-cover photo of the Nya Humanisten building in Gothenburg
by Jessica Oscarsson, University of Gothenburg.

Published by:
Linköping University Electronic Press, Sweden
Linköping Electronic Conference Proceedings, No. 184
NEALT Proceedings Series, No. 55

ISBN: 978-91-7929-614-8
ISSN: 1736-8197
eISSN: 1736-6305
ISSN: 1650-3686
eISSN: 1650-3740

# Preface

SLTC-2020, the 8th Swedish Language Technology Conference, took place 25–27 November 2020. As it is tradition for SLTC, the first day was reserved for workshops, and the main conference took place during the last two days. Contrary to tradition, the conference was held completely online, but had there not been a Covid-19 pandemic we would have been at the University of Gothenburg, in the newly renovated Humanisten building.

There were four associated SLTC workshops, all taking place 25 November:

- NLP4CALL: NLP for Computer-Assisted Language Learning
- RESOURCEFUL: Resources and Representations for Under-Resourced Languages and Domains
- Computational Detection of Language Change
- Applied Swedish NLP

For the main conference, 26–27 November, there were 193 registered participants, of which 60% were from Sweden and 40% came from 33 different countries. Of the registered participants, between 40–60 showed up at each session.

In total there were 33 presentations in 11 sessions, of which three invited keynote talks by Leon Derczynski, Vera Demberg, and Raquel Fernández.

One of the main ideas with SLTC is that it should be a light-weight conference, with the main focus being on discussions, networking, and socialising. We encourage late-breaking reports on unfinished work, as well as presentations of emerging research ideas and "side projects". Therefore, we only allow short paper submissions of at most 4 pages, and we do not publish any proceedings.

But there is always interest from some of the authors to publish a longer version of their SLTC presentation. Enter the SLTC post-proceedings!

Directly after the conference, we invited all authors who presented at SLTC-2020 to submit a longer post-proceedings version of their work. The initial submission deadline was at the end of February 2021, and after peer review followed by revisions, all papers were ready for publishing at the end of June 2021.

We are proud to present the following nine papers, all of which are longer, enhanced versions of presentations from SLTC-2020:

- David Sabiiti Bamutura: *Ry/Rk-Lex: A Computational Lexicon for Runyankore and Rukiga Languages*
- Dana Dannélls and Shafqat Virk: *A Supervised Machine Learning Approach for Post-OCR Error Detection for Historical Text*
- Yaroslav Getman: *Automated Writing Support for Swedish Learners*
- Harald Hammarström, One-Soon Her and Marc Tang: *Term Spotting: A Quick-and-dirty Method for Extracting Typological Features of Language from Grammatical Descriptions*
- Oskar Jerdhaf, Marina Santini, Peter Lundberg, Anette Karlsson and Arne Jönsson: *Implant Term Extraction from Swedish Medical Records – Phase 1: Lessons Learned*
- Maryam Rajestari, Simon Dobnik, Robin Cooper and Aram Karimi: *Very Necessary: The Meaning of Non-gradable Modal Adjectives in Discourse Contexts*

- Jonas Sjöbergh and Viggo Kann: *Granska API – an Online API for Grammar Checking and Other NLP Services*
- Søren Wichmann: *Pipeline for a Data-driven Network of Linguistic Terms*
- Niklas Zechner: *Cross-Topic Author Identification – a Case Study on Swedish Literature*

### Acknowledgements

We would like to thank everyone who made this possible: the authors of the papers in this volume, as well as all authors and presenters at the SLTC-2020 conference; the organisers, helpers, session chairs, and keynote speakers of the SLTC-2020 conference; and finally NEALT, the Northern European Association for Language Technology.

Peter Ljunglöf, Simon Dobnik, and Richard Johansson
Editors
Gothenburg
June, 2021

# Table of Contents

# Ry/Rk-Lex: A Computational Lexicon for Runyankore and Rukiga Languages

**David Sabiiti Bamutura**
Chalmers University of Technology / Gothenburg, Sweden
Mbarara University of Science & Technology / Mbarara, Uganda
bamutra@chalmers.se | dbamutura@must.ac.ug

## Abstract

Current research in computational linguistics and NLP requires the existence of language resources. Whereas these resources are available for only a few well-resourced languages, there are many languages that have been neglected. Among the neglected and / or under-resourced languages are Runyankore and Rukiga (henceforth referred to as *Ry/Rk*). In this paper, we report on *Ry/Rk-Lex*, a moderately large computational lexicon for Ry/Rk that we constructed from various existing data sources. Ry/Rk are two under-resourced Bantu languages with virtually no computational resources. About 9,400 lemmata have been entered so far. Ry/Rk-Lex has been enriched with syntactic and lexical semantic features, with the intent of providing a reference computational lexicon for Ry/Rk in other NLP (1) tasks such as: morphological analysis and generation; part of speech (POS) tagging; named entity recognition (NER); and (2) applications such as: spell and grammar checking; and cross-lingual information retrieval (CLIR). We have used Ry/Rk-Lex to dramatically increase the lexical coverage of previously developed computational resource grammars for Ry/Rk.

## 1 Introduction

Almost all computational linguistics and natural language processing (NLP) research areas require the use of computational language resources. However, such resources are available for a few well-resourced and ”politically advantaged” languages of the world. As a result, most languages remain neglected. Recently, the NLP community has started to acknowledge that resources for under-resourced languages should also be given priority. Why? One reason being that as far as language typology is concerned, the few well-resourced languages do not represent the structural diversity of the remaining languages (Bender, 2013).

This study is a follow-up to a previous, but related study on the engineering of computational resource grammars for Runyankore and Rukiga (henceforth referred to as *Ry/Rk*) (Bamutura et al., 2020), using the Grammatical Framework (GF) and its Resource Grammar Library (Ranta, 2009a,b). In the previous study, a narrow-coverage lexicon of 167 lexical items was sufficient for grammar development. In order to both encourage wide use of the grammar (in real-life NLP applications) and fill the need for computational lexical language resources for Ry/Rk, it was necessary to develop a general-purpose lexicon. Consequently, we set out to create *Ry/Rk-Lex*, a computational lexical resource for Ry/Rk. Despite the challenges faced due to lack of substantial open source language resources for Ry/Rk, we have so far entered about 9,400 lemmata into Ry/Rk-Lex. Ry/Rk has been enriched with syntactic and lexical semantic features, with the intent of providing a reference computational lexicon for Ry/Rk that can be used in other NLP tasks and applications.

### 1.1 Runyankore and Rukiga Languages

Ry/Rk are two languages spoken by about 3.4 and 2.4 million people (Simons and Fennig, 2018) respectively. They belong to the *JE10* zone (Maho, 2009) of the Great Lakes, Narrow Bantu of the Niger-Congo language family. The native speakers of these languages are called Banyankore and Bakiga respectively. The two peoples hail from and / or live in the regions of Ankole and Kigezi — both located in South Western Uganda, East Africa.

Just like other Eastern Great Lakes Bantu languages, Ry/Rk are *mildly tonal* (Muzale, 1998), *highly agglutinating* with a *large noun class system* (Katushemererwe and Hanneforth, 2010; Byamugisha et al., 2016). They exhibit high incidences of *phonological conditioning* (Katushemererwe et al., 2020) that makes them complex to deal with computationally. The agglutinating nature, intricate concordial agreement system and phonological conditioning make it more difficult to model and formalise the grammars for these languages using the symbolic approach. For details about the nominal and verbal morphology of these languages from the perspective of computational linguistics, the reader should see (Katushemererwe, 2013; Byamugisha, 2019; Bamutura et al., 2020; Katushemererwe et al., 2020).

### 1.2 Challenges of Creating Computational Lexica for Runyankore and Rukiga

Though Ry/Rk languages are spoken by a sizeable population they are under-resourced and have a limited presence on the web. When we consider the creation of computational language resources for these languages, four major problems stand out: (1) large amounts of language data must be collected manually by copy-typing which is time-consuming and error-prone; (2) refusal by publishers of books and dictionaries to allow their texts to be used as sources of these data; (3) lack of an easy to use and extensible modelling and storage format for computational lexicons for Bantu languages; and (4) lack of funds to procure copyrighted works for the extraction and processing of computational lexicons and other resources. These lexical resources are however very important for the success of other NLP (1) tasks such as: morphological analysis and generation; part of speech (POS) tagging; named entity recognition (NER); and (2) applications such as spell and grammar checking ; and cross-lingual information retrieval (CLIR).

### 1.3 Research Questions

This study was guided by the following research questions:

**RQ.1** What are the existing linguistic data sources that can be used for the development of computational lexicons for Ry/Rk?

**RQ.2** Out of the sources identified in RQ.1, which sources are suitable for use as a computational lexicon for Ry/Rk?

**RQ.3** How can computational lexicons for Ry/Rk be extracted and modelled or structured in a simple, flexible and extensible manner?

The rest of the paper is structured as follows: Section 2 presents related work; Section 3 presents the data used for the study, its sources, how it was curated and processed; and Section 4 presents the findings in terms of how Ry/Rk-Lex was described i.e. how the different parts of speech were handled, the persistence structure that was used for storage of lexical items. Results & discussion are presented in Section 5. Lastly, Section 6 presents conclusion and future work.

## 2 Related Work

### 2.1 Computational Lexica

Machine Readable Dictionaries (MRDs) and computational lexicons for well-resourced languages such as those reported by Sanfilippo (1994), and ACQUILEX projects I and II[1] were created from existing conventional dictionaries. The aim in those studies was to explore lexical language analysis use cases such as building lexical knowledge-bases. The task of creating MRDs was made easier because the dictionaries used had machine-readable versions that were made available i.e. without copyright restrictions.

In the case of Ry/Rk, such an approach is difficult largely because Ry/Rk dictionaries do not include rich morphosyntax (mainly due to the complex morphology). Additionally, most of the dictionaries are protected by copyright. The lexical semantic relation information (hypernymy and meronymy) provided in the Runyankore and Rukiga thesaurus (Museveni et al., 2012) would be a good starting point but it is also copyrighted.

In addition to having MRDs, well-resourced languages possess the following: large amounts of language data available on the web; prepared corpora of good quality; treebanks (Xiao, 2008; Taylor et al., 2003; Böhmová et al., 2003); and lexical databases such as the original English WordNet (Miller, 1995) and subsequent additions (Christiane and Miller, 1998). Petrolito and Bond (2014) provide a comprehensive survey of different existing language-specific WordNet-based lexical databases and Navigli and Ponzetto (2010) describe a wide-coverage multilingual semantic network derived from combining WordNet and Wikipedia. These resources make the creation of computational lexical resources easier for these languages. It is important to note that the same resources were developed by well-funded research groups.

Among the Bantu languages, computational lexicons have been developed for some languages such as Swahili (Hurskainen, 2004) in East Africa, and isiZulu and isiXhosa (Bosch et al., 2006) in South Africa using XML and related technologies for

[1] see: `https://www.cl.cam.ac.uk/research/nl/acquilex/`

modelling and annotation. The computational lexicon for Swahili — developed as part of the Swahili Language Manager (SALAMA) — and other South African languages are perhaps the most comprehensive in terms of: (1) the number of lexical items covered and (2) addressing lexical semantic relation issues such as synonymy. The lexical resource for South Africa has been expanded (both by size and number of languages) and converted into the African WordNet (AfWN) to include other southern Africa Bantu languages namely; Setswana, Sesotho, isiNdebele, Xitsonga and Siswati (Griesel and Bosch, 2014, 2020). However, there has been no attempt to create an enriched computational lexical resource for Ry/Rk.

### 2.2 Computational Lexicon Modelling

With regard to modelling of lexicons for Bantu languages, a Bantu Language Model (BantuLM) was put forward by Bosch et al. (2006, 2018) after eliciting the inadequacies of Lexical Markup Framework (LMF) (Francopoulo et al., 2006) arising from a failure to take such morphologies into account when designing the framework. It was also posited that using BantuLM to prepare lexical resources would encourage cross-language use cases. Bosch et al. (2006) implemented BantuLM using XML and related technologies, while Bosch et al. (2018) switched to an ontology-based approach for describing lexicographic data that combined the best of the Lexicon Model for Ontologies and the Multilingual Morpheme Core Ontology (MMoOnCore) to realise the features envisaged in the BantuLM. Although ontology-based methods encourage the cross-linking of multilingual data, they require a knowledge-base of lexical semantic relations. With the exception of synonym information available in some dictionaries (Taylor and Mapirwe, 2009; Mpairwe and Kahangi, 2013a; Museveni et al., 2009) and basic semantic relations found in a thesaurus for Ry/Rk (Museveni et al., 2012), there are no other sources for such data. Use of ontology-based (semantic networks) for lexical language resources necessitates the formalising the meaning of lexical items beyond word definitions (also called glosses) which current sources do not provide. Going beyond definitions or glosses requires a separate study with huge human and capital resources to turn these resources into lexical semantic networks such as WordNet. YAML[2] was chosen for the preparation, storage and sharing of the Ry/Rk lexicon because for our current purposes we do not require the complex modelling provided for by BantuLM.

## 3 Data Sources, Curation & Processing

### 3.1 Existing Data Sources

In total, fourteen linguistic data sources summarised in Table 1 were identified (by web-search, visiting bookshops and publishing houses in Uganda) as the existing data sources that could be used for the development of electronic corpora and or lexica for Ry/Rk. Due to copyright restrictions, we used five of the fourteen sources in whole for lexical resource creation. These five sources (identified as; RRDict1959, RRBibleNew1964, RRSCAWL2004, RRUDofHR and RREthics) are marked using * in that table. However, as explained later in detail in section 3.2.4, we used RRNews2013-2014 (marked with † in the same Table 1) in whole but have made deliberate effort to make sure that only small random fragments of the corpus can be released for demonstration purposes in an academic setting. Other sources marked with ‡ were used solely for reference in case of lack of knowledge.

### 3.2 Data Curation & Processing

Having obtained sources of data that could be used, the language data contained in those sources had to be extracted and pre-processed in order to obtain individual word tokens. Because the methods used were slightly different for each data source, we explain the process used for each in Sections; 3.2.3, 3.2.1, 3.2.2 and 3.2.4. The procedures used for RRUDofHR and RREthics are identical to those described in section 3.2.2 and 3.2.4 respectively because the former was also scraped from the web while the later required scanning of a hard copy.

#### 3.2.1 RRDict1959

To the best of our knowledge, there is only one MRD for Ry/Rk identified as RRDict1959 in Table 1. It was extracted from the dictionary by Taylor (1959). The MRD is freely available for use as long as one abides by a Bantuist Manifesto.[3] On close inspection of the entries, a number of anomalies were found: (1) singular and plural forms of nouns are entered as separate entries, (2) some entries do

[2] A markup language available at: https://yaml.org

[3] The manifesto can be read at http://www.cbold.ish-lyon.cnrs.fr/Docs/manifesto.html

not qualify as lemmata because they possess additional and unnecessary derivational and inflectional morphemes, (3) lack of conjugation information for verbs, (4) lack of new lemmata that have been introduced to Ry/Rk since 1959, and (5) entries lack synonym information. The first three anomalies were corrected manually by eliminating non-lemma entries, stripping off the unnecessary affixes and providing verbal morpheme endings that guide verb conjugation. For example, we did not agree with the use of the /*ku*/ morpheme as a prefix before a verb because it is unnecessary. Placing /*ku*/ before the verb is akin to placing the word /*to*/ before every verb in English and yet /*to*/ is rarely entered in dictionaries. It is also an unnecessary repetition. The same was done during lemmatisation of verbs from other sources.

### 3.2.2 RRBibleNew1964

Since a digital version of the New Testament Bible in Runyankore-Rukiga (identified as RRBibleNew1964 in Table 1) is available, it was scrapped from the web after which text pre-processing was done. This pre-processing included text cleaning (removal of HTML markup text, chapter and verse identifiers), text tokenisation, lemmatisation, POS tagging and annotation of each lexical item with simple inflectional morphology i.e. conjugation for verbs, noun class information for nouns, definition glosses for English and synonyms. Lemmatisation and POS tagging were done manually by 4 research assistants. For lemmatisation of verbs, we chose to use the radical concatenated with a final morpheme which most of the time is simply a vowel, called the Final Vowel (FV). This final morpheme is the verbal ending used for the experiential present tense. The open-source machine readable dictionary (RRDict1959) was used to validate our lemmatisation, POS tagging and noun-class identification process for words that existed in the dictionary.

### 3.2.3 RRSCAWL2004

RRSCAWL2004 is an English–French bilingual list of 1,700 words that was compiled and suggested by Snider and Roberts. (2004) as a useful seed-list for any researcher doing comparative linguistic studies on African languages. Because this list was prepared for Africa, it is highly likely to capture the common concepts used by the ordinary African, such as Ry/Rk speakers. The words in the list are organised semantically under twelve main headings with further subdivisions. The words cover concepts ranging from human to non-human and from concrete to abstract. Since the data is presented within tables of a file in PDF, we used Tabula,[4] a piece of free software to quickly extract these tables locked up in PDF. Tabula is able to export that data into comma separated values (CSV) or Microsoft Office Excel file formats. We hired a professional translator to translate the English glosses to Runyankore and Rukiga. The resulting list was further annotated and fed into Ry/Rk-Lex.

### 3.2.4 RRNews2013-2014

From scanned images of Orumuri Newspaper, we used the Optical Character Recognition (OCR) feature for English found in Adobe Acrobat Pro DC[5] to extract text from the images. This text was copied and pasted in xml documents that served partially to preserve the structure and content of the newspaper and its articles. Due to the lack of existing OCR software trained specifically on Ry/Rk, errors were encountered and these were corrected manually. Sometimes, it required copying sentence by sentence or paragraph by paragraph. There were two major types of errors: simple spelling mistakes and unrecognisable characters spanning one or several lines of an article. The line errors were mainly associated with Ry/Rk words that contained /ii/ or /aa/ and we are still investigating the reason(s) for this behaviour. Other problems emanated from lists illustrated using bullet points. We used xml to divide the structure of the newspaper into several sections: (1) Amakuru, (2) Amabaruha, (3) Amagara, (4) Shwenkazi, (5) Regional News (Kigezi, Bushenyi, Mabara), (6) Omwekambi and (7) Emizaano. Although the news corpus collected is of poor quality in terms of grammar (Katushemereirwe, personal communication), it is lexically rich and contains words that have been introduced in the languages due to interaction with other languages and globalisation. It therefore contributes significantly to the number of words used currently in contemporary Ry/Rk that are not contained in RRDict1959, RRBibleNew1964, RRVoc2004 and RRSCAWL2004. RRNews2013-2014 was cleaned, tokenised and lemmatised in the same way as RRBibleNew1964 as described in 3.2.2 above.

[4] See: `https://tabula.technology/`

[5] Version: 221.001.20145 for Mac OS X

### 3.3 Summing It Up

After pre-processing RRDict1959 to remove the first three anomalies mentioned previously in section 3.2.1, the data obtained was used to validate our lemmatisation, POS tagging and noun-class identification process for lemmata that exist in both RRDict1959 and those that were manually extracted from the completed parts of RRBibleNew1964, RRUDofHR, RREthics, RRSCAWL2004 and RRNews2013–2014. Since text from RRDict1959 and RRBibleNew1964 is dated, the lemmata obtained from the manually created corpus from Orumuri,[6] a weekly Runyankore-Rukiga newspaper, RRUDofHR, RREthics, and lemmata obtained from RRSCAWL2004 and RRVoc2004 (Kaji, 2004) were used to update the RyRk-Lex with words currently used in RyRk. It should be noted that the creation of the RRCorpus and its processing for lexicon extraction is still ongoing.

## 4 Findings: Ry/Rk-Lex Description

The properties or features for each lemma depend on a number of factors but the major determinants are: the part of speech (POS); the language to which the lemma belongs; and availability of synonyms and definition glosses in English. While the language property is mandatory for all lemma entries, verbs present a problem because the lemma is usually identical for both languages but its method of conjugation differs for each language. We kept the field mandatory for the simple reason that the lemma belongs to both languages although conjugated differently by each language as explained with an example in Subsection 4.2. Otherwise, the properties peculiar to each part of speech are discussed in the subsections below. These properties are illustrated in Table 2 which summarises the structure of Ry/Rk-Lex as specified in a schema[7] we developed whose structure is further described in Section 4.1.

### 4.1 Ry/Rk-Lex Persistence Structure

For purposes of preparing a shareable resource, we described and stored each entry using YAML. Entries are entered according to a YAML Schema that we designed. Ry/Rk-Lex is shareable because of the schema which communicates the structure of the lexicon. The schema was also utilised for validation of Ry/Rk-Lex in order to identify and correct errors. Manually identified synonyms have been entered for some lemma entries in Ry/Rk-Lex but have not yet been cross-linked.

### 4.2 Verbs

We have obtained, prepared and stored about 3500 verbs. The verbal features covered include the lemma which is the radical[8] and its final vowel for the experiential present tense (Muzale, 1998; Bamutura et al., 2020). The entry is complemented by a conjugation field that demonstrates how the verb can be conjugated to any of the tenses in Ry/Rk i.e. far past, near past, experiential present, memorial present, near future and far future. Interestingly, the key to performing that conjugation correctly depends on knowing the morpheme for the perfective aspect for the post radical position of the verb. This morpheme is allomorphic and therefor realised differently. The allomorph chosen for a particular verb depends on the following four properties of the verb in experiential present: (1) the syllable structure (2) the penultimate vowel, (3) length of the penultimate vowel and (4) terminal syllable of the verb (Mpairwe and Kahangi, 2013b). Mpairwe and Kahangi (2013b) further attempt at describing these rules for determining the allormorphs as a rule-based procedure or "pseudo" algorithm. Although these rules are natural to a native speaker of the languages, attempts at implementing them as a computer program produced sub-optimal results. .

The verb type field specifies the valency of the verb ignoring any valency increasing derivational suffixes i.e extensions for applicative and causative constructions. Since this lexicon covers two closely related languages, each lemma belonging to the verb POS is annotated with a property for specifying the language. As already mentioned previously, the value for the language field does not depend only on the radical or stem but also the way the verb is conjugated. For instance the verb /reeta/ meaning /bring/ would be conjugated to /reet + sire/ and /ree + sire/ resulting in the surface forms /reetsire/ and /reesire/ in the perfective aspect for Runyankore and Rukiga respectively. Therefore the conjugation field for verbs could be put at top level node but to be more specific it should appear under the conjugation node. We decided to do it at

[6] The publisher, Vision Group terminated the publication of the newspaper in 2020

[7] See appendix I for the full structure

[8] A radical is a sub unit of a stem taken from the base, for details, see Meeussen (1967)

| Source | ID | type/Genre | mode | copyright |
|---|---|---|---|---|
| Taylor (1959) | RRDict1959* | Dictionary | MRD | Free |
| New Testament Ry/Rk Bible | RRBibleNew1964* | Religion | electronic | Free |
| Snider and Roberts. (2004) | RRSCAWL2004* | Word List | PDF | Free |
| Taylor and Mapirwe (2009) | RRDict2009 | Dictionary | hard copy | restricted |
| Kaji (2004) | RRVoc2004‡ | Vocabulary List | hard copy | restricted |
| Orumuri | RRNews2013-2014† | Newspaper | hard copy | restricted |
| Morris and Kirwan (1972) | RRGrammar1972‡ | Grammar book | hard copy | restricted |
| Mpairwe and Kahangi (2013b) | RRGrammar2013‡ | Grammar book | hard copy | restricted |
| Mpairwe and Kahangi (2013a) | RRDict2013 | Dictionary | hard copy | restricted |
| Museveni et al. (2009) | RRDict2009 | Dictionary | hard copy | restricted |
| Museveni et al. (2012) | RRThes2012 | Thesaurus | hard copy | restricted |
| Karwemera (1994) | RRCgg1994 | Book | hard copy | restricted |
| Universal Declaration of Human Rights | RRUDofHR* | Law | electronic | free |
| Government communication | RREthics* | Simplified law | hardcopy | free |

Table 1: Summary of data sources for corpora and lexical resources. Note: Items marked with * were used without special consideration of copyright. Those with † were used in whole but the resulting corpus will unfortunately not be freely available. Those with ‡ were used solely for reference i.e. lookup of particular information such as synonyms and lemmas for closed categories.

| property | type | Optionality | Description |
|---|---|---|---|
| lemma | string | Mandatory | The conventional citation form of a lexical item |
| lemma_id | integer | Mandatory | The numerical identifier of the lemma |
| pos | map | Mandatory | The part of speech defined at two levels of granularity. |
| eng_defn | string | Mandatory | A definition of the lemma in English |
| synonyms | sequence | Mandatory | A list of synonyms for the lemma |
| lang | sequence | Mandatory | A list of language identifiers for the lemma |
| conjugations | sequence of maps | Optional | Non-perfective and perfective Verbal-endings |
| noun_class | sequence of strings | Optional | Noun class information for nouns |

Table 2: Top-level properties for each lemma entry in Ry/Rk-Lex. Each property in column one has a type provided in column two. Column three indicates whether the property is mandatory or optional for each lemma entry while the last column provides a description of the property.

|  | NC | NCP | Individual Particles |  | Example |  | Gloss |
|---|---|---|---|---|---|---|---|
| ID | Numbers | Particles | Singular | Plural | Singular | Plural | Singular(Plural) |
| 1 | $\beta$ | ZERO_N | n/a | N | n/a | embabazi | n/a (mercy / mercies) |
| 2 | $\sigma$ | N_ZERO | N | n/a | enzigu | n/a | vengeance (n/a) |
| 3 | $\gamma$ | RU_ZERO | RU | n/a | 0-ru-me | n/a | dew (n/a) |

Table 3: Examples of Ry/Rk nouns without noun classes (NC). Their associated noun class particle (NCP) pairs are shown but the equivalent numeric identifiers as used by the Bleek-Meinhoff system of numbering could not be identified. We therefore used greek letters to represent the unknown.

| Part-of-Speech | # of lemmata |
|---|---|
| Verbs | 3532 |
| Common Nouns | 4789 |
| Proper Nouns | 523 |
| Determiners | 124 |
| Pronominal Expressions | 85 |
| Adverbs | 140 |
| Prepositions | 43 |
| Adjectives | 148 |
| Conjunctions & Subjunctions | 45 |
| Total | 9429 |

Table 4: Number of entries made in Ry/Rk-Lex for each part of speech.

both levels, in order to recognise that the lemma is for both Rukiga and Runyankore but demand any developed parser to further crosscheck for the language property under conjugation.

### 4.3 Common Nouns and Proper Nouns

In addition to all properties considered mandatory, noun class information was added as an additional field. Both numerical noun classes and textual noun class particles are provided. During lexical collection and processing, three additional categories of nouns that do not fit in the conventional noun class system for Ry/Rk used by Katushemererwe and Hanneforth (2010); Turyamwomwe (2011); Byamugisha et al. (2016) were encountered. An example from each category is illustrated in Table 3.

### 4.4 Nominal Qualificatives

Nominal qualificatives are expressions that usually qualify nouns, pronouns and noun phrases, and in Ry/Rk include (1) adjectives, (2) adjectival stems and phrases, (3) nouns that qualify other nouns (4) enumeratives (both inclusive and exclusive), (4) relative subject clauses and (5) relative object clauses (Mpairwe and Kahangi, 2013b). Only the nominal qualificatives (1)–(3) were included. Qualificatives (4) and (5) were excluded because they are clauses. Mpairwe and Kahangi (2013b) mention in their grammar book that the notion of adjectives as understood in English results in limited number of adjectives when applied to Ry/Rk. The adjectives are not more than twenty in number. There are however other ways of expressing qualification of nominal expressions in Ry/Rk. We therefore found it difficult to identify and classify all forms of this part-of-speech. In addition to the mandatory properties, four additional properties were required to have adjectives and other nominal qualificatives adequately described. The properties included: position (whether the adjective is located before or after the noun), doesAgree (which indicates whether the adjective changes with respect to the noun class of the nominal being modified), and isProper (a boolean field that captures whether the adjective is a stand-alone or one that requires modification by a suffix). Some adjectival expressions are multi-word expressions (portmateau) such as clauses. These clauses are usually derivational and therefore have been left out of the lexicon.

### 4.5 Adverbs and Adverbial expressions

Both Schachter and Shopen (2007) and Cheng and Downing (2014) define the adverb as that part-of-speech that modifies all other parts-of-speech apart from the noun. The Universal Dependencies (UD)[9] provides a more concrete definition i.e. “adverbs are words that typically modify verbs for categories such as time, place, direction or manner and they may also modify adjectives and other adverbs”. The single exclusion of nouns by all definitions implies that this part of speech is an amalgamation of different words, phrases and clauses as long as they do not modify nouns or noun phrases. For Ry/Rk, Mpairwe and Kahangi (2013b) define it as a word, phrase or clause that answers questions based on the question-words: *where* (for adverbs of place), *when* (for adverbs of time, frequency and condition), *how* (for adverbs of manner and comparison), and lastly *why* (for adverbs of reason or purpose and concession). Most adverbials in Ry/Rk are a single word consisting of two or more words when translated to English. In other words you have a single-word consisting of two or more morphemes belonging to multiple parts of speech. A good example is the word /*kisyo*/ which means /*like that*/ in English and belongs to singular forms of nouns from noun classes 7_8. The associated word /*bisyo*/ for the plural form implies that the stem is /*syo*/. In describing or extracting lemmata for adverbs, we concentrated on adverbial expressions that were easily discernible from a single word. We advise that further work be done for adverbials especially those that span multiple words by obtaining them from professionally annotated corpora alongside detailed annotation guidelines. For instance the multi-morpheme words could obtained from a Ry/Rk corpus that has been anno-

[9] See:https://universaldependencies.org/u/pos/ADV.html

tated using annotation guidelines that are based on a more linguistically sound theory for word class division for Ry/Rk.

### 4.6 Closed Categories

POS that belong to the closed category are generally few but occur frequently in a corpus. Whereas conjunctions (including subjunctions), prepositions, determiners and quantifiers are actually few in number for Ry/Rk, pronouns constitute a large number. Notably, most POS from the closed category can be adquately covered by working through grammar books such as; (Morris and Kirwan, 1972), (Taylor and Mapirwe, 2009), (Mpairwe and Kahangi, 2013b) and (Ndoleriire, 2020).

#### 4.6.1 Pronouns

Generally, pronouns are words that substitute for nouns or noun phrases and whose meaning is recoverable through anaphora resolution sometimes requiring investigation of linguistic context beyond the sentence. In Ry/Rk, pronominal expressions are either single-word expressions (called pronouns) or pronominal affixes (morphemes) (Mpairwe and Kahangi, 2013b; Katushemererwe et al., 2020). Manually identifying and annotating a single-word pronoun from a tokenised corpus whose sorting is based on most frequent word is much easier than doing the same for pronominal affixes because you lose contextual information that would help with identification. We therefore decided to concentrate on discrete pronouns.

Otherwise, in order to describe and use self-standing or independent pronouns, terms used by (Mpairwe and Kahangi, 2013b,a) and (Katushemererwe et al., 2020) respectively to refer to those pronouns that do not require to be affixed to another POS, the parameters: grammatical gender (noun class), number, person and type of pronoun are required and were captured for this particular POS. Those that have not been covered are affix-based pronouns.

## 5 Reflections and Discussion

At the time of writing, Ry/Rk-Lex currently consists of 9,429 lemmata of various parts-of-speech summarised in Table 4. From the breakdown we note that verbs and nouns make up the largest share of the total number of lemmata. For the case of verbs, the large number is attributed to the fact that new verbs can be formed via derivation processes such as reduplication, reciprocation and in some cases through the use of applicative and causative constructions common among Bantu languages. Nouns are inherently numerous since they name things. Deverbatives have been excluded so far from Ry/Rk-Lex because they are easy to add once all verbs are known. Despite the low number of proper nouns in Ry/Rk-Lex, this category of nouns is huge and we plan to add more from the Ry/Rk Thesaurus (RRThes2012) after obtaining copyright permission. In Ry/Rk, adverbs are a complicated part of speech. They mostly exist as adverbial expressions constructed from locative noun class particles: */mu/*, */ku/* and */ha/*. As a result, only a few have been considered as lemmata so far but more will be included in future. Parts of speech that belong to closed categories are few and consist of the most frequently used words. For each lemma, we tried our best to enter as much synonym information as we could. However, cross-linking of synonyms has not yet been done due to time constraints but we plan to do it in future. We manually fixed and updated each entry with more information specifically conjugation for verbs and correct noun classes for nouns.

While processing nouns, nouns that did not fall under the accepted noun class numerical system were encountered. In Table 3, examples of such nouns are provided. We suggest that the noun classes used in the numeral system be expanded as some nominal lexical items cannot be brought under the pre-existing numerical system used in literature for Runyankore-Rukiga. Since the notion of adjectives and or nominal qualifiers in Ry/Rk is very limited as mentioned before in subsection 4.4, we found it difficult to identify and classify all forms of this part of speech.

For each lemma entered in the lexicon, a language field is provided to indicate the language the lemma belongs to. A lemma that is used by both languages is annotated with *’all’* while ISO 693-3 three-letter codes *‘nyn’* and *‘cgg’* are utilised to annotate lemmata that are exclusively used by either Runyankore or Rukiga respectively. It is therefore possible to to automatically extract particular parts of the lexicon for each language. Ry/Rk-Lex attempts to provide a definition in the English language for each lemma despite the fact that this approach to lexical semantics suffers from a number of problems, one of which is circular definitions.

Any current work on lexical resources would expect the inclusion of lexical semantic relations (synonymy, hypernymy and meronymy) within the resource. Though we have provided some synonym information in Ry/Rk-Lex, we have not yet cross-linked the synonyms. Since YAML provides anchors and references as features, they can be exploited to link synonyms together. Hypernymy and meronymy relations can also be included using a similar method provided knowledge and monetary resources are made available. Since building and maintaining a lexicon is a never-ending process, we are continuously updating it with lemmata as we find more texts written in the language or using free word lists such as: The SPECIALIST LEXICON[10] (Browne et al., 2018); and or the lexicon embedded in the SimpleNLG API and the English Open Word List (EOWL)[11] prepared by Loge (2015). It contains 128,985 words and was extracted from the UK Advanced Cryptics Dictionary (UKACD) Version 1.6.

## 6 Conclusion and Future Work

In this paper, we have described the creation of Ry/Rk-Lex, a computational lexicon for Ry/Rk. It currently consists of 9,429 lemma entries. Since the languages are under-resourced, we found only fourteen data sources that could be used for its creation. Of the fourteen, only five were utilised as a whole without special consideration of violation of copyright because they are free from copyright. In order to store and make the resource shareable, we designed a schema for structuring the lexicon and used it to organise and annotate all lemmata that have been extracted from the data sources by both manual and automatic methods.

As future work, we plan to build and evaluate conjugation, lemmatisation, morphological analyser and generator, POS tagging software for Ry/Rk that can be used to speed up the process of language resource creation. With these software tools in place, Ry/Rk-Lex can also be used for developing systems for cross-lingual information retrieval (CLIR) especially for people with moderate to poor competence in English but competent in writing Ry/Rk.

For a broader audience, the CLIR system could be augmented with an automatic speech recognition (ASR) module for Ry/Rk targeted towards specific domains. Although Ry/Rk-Lex does not contain all lexical semantic knowledge, our resource can still be used as a starting point for the computational formalisation of the lexical semantics of Ry/Rk and for developing an Ry/Rk WordNet. In its current form, Ry/Rk-Lex has been used to dramatically improve (from 167 to 9,429 lemmata) the lexical coverage of the computational resource grammars of Ry/Rk.

Lastly, there is also need to do more research on establishing a linguistically motivated and sound theory or criteria for word class division and / or drawing the thin line between morphology and lexicon for Ry/Rk as a Bantu language. Using such a criteria would result into lexica that does not appear to be modelled on English and or Latin-based languages. For Ry/Rk-Lex, the word class division was inspired by Indo-European languages and used by GF. However, establishment of a common ground amongst languages in the tradition of the Universal POS tags[12] and the general guidelines put forward by UD version 2 project on the handling of morphology[13] is currently the main focus and future direction this research study.

## Acknowledgments

The author would like to acknowledge and thank his academic advisers: Peter Ljunglöf and Peter Nabende for their insightful comments and reviews, research assistants; Ms. Fulgencia Kabagyenzi, Ms. Juliet Nsimire, Ms. Doreck Nduhukire, Mr. Anyine Cranmar for their tireless work in transcribing some of the data sources, and Mr. Jackson Ndyanimanya for translating the RRSAWL2004.

Special thanks to the two reviewers who gave very detailed and helpful comments that have to a great extent improved the quality manuscript.

This work was supported by the Sida / BRIGHT Project 317 under the Makerere-Sweden Bilateral Research Programme 2015–2020 and 2020–2021.

## References

David Bamutura, Peter Ljunglöf, and Peter Nebende. 2020. Towards Computational Resource Grammars for Runyankore and Rukiga. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2846–2854, Marseille, France. European Language Resources Association.

[10] Available at https://lexsrv3.nlm.nih.gov/LexSysGroup/Projects/lexicon/current/web/release/2020.html

[11] see: https://diginoodles.com/projects/eowl

[12] See:https://universaldependencies.org/v2/postags.html

[13] See:https://universaldependencies.org/u/overview/morphology.html

Emily M. Bender. 2013. *Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The Prague Dependency Treebank. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 103–127. Springer Netherlands, Dordrecht.

Sonja Bosch, Thomas Eckart, Bettina Klimek, Dirk Goldhahn, and Uwe Quasthoff. 2018. Preparation and usage of Xhosa lexicographical data for a multilingual, federated environment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Sonja E. Bosch, Laurette Pretorius, and Jackie Jones. 2006. Towards machine-readable lexicons for south African Bantu languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Allen C. Browne, Alexa T. McCray, and Suresh Srinivasan. 2018. The SPECIALIST LEXICON. Technical report, Lister Hill National Center for Biomedical Communications, National Library of Medicine, Beshesda, Maryland.

Joan Byamugisha. 2019. *Computational Morphology and Bantu Language Learning: An Implementation for Runyakitara*. PhD Dissertation, University of Cape Town, Computer Science Department.

Joan Byamugisha, C. Maria Keet, and Brian DeRenzi. 2016. Bootstrapping a Runyankore CNL from an isiZulu CNL. In *Controlled Natural Language*, pages 25–36. Springer International Publishing.

Lisa Cheng and Laura Downing. 2014. *The problems of adverbs in Zulu*, pages 42–59. John Benjamins Publishing Company.

Fellbaum Christiane and George A. Miller, editors. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Gil Francopoulo, Monte George, Nicoletta Calzolari, Monica Monachini, Nuria Bel, Mandy Pet, and Claudia Soria. 2006. Lexical markup framework (LMF). In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Marissa Griesel and Sonja Bosch. 2014. Taking stock of the African Wordnet project: 5 years of development. In *Proceedings of the Seventh Global Wordnet Conference*, pages 148–153, Tartu, Estonia. University of Tartu Press.

Marissa Griesel and Sonja Bosch. 2020. Navigating challenges of multilingual resource development for under-resourced languages: The case of the African Wordnet project. In *Proceedings of the first workshop on Resources for African Indigenous Languages*, pages 45–50, Marseille, France. European Language Resources Association (ELRA).

Arvi Hurskainen. 2004. Swahili language manager: A storehouse for developing multiple computational applications. *Nordic Journal of African Studies*, 13(3):363 – 397.

Shigeki Kaji. 2004. *A Runyankore Vocabulary*. Research Institute for Languages and Cultures of Asia and Africa (ILCAA), Tokyo University of Foreign Studies in English.

Festo Karwemera. 1994. *Emicwe n'Emigyenzo y'Abakiga*. Fountain Publishers, Kampala, Uganda.

Fridah Katushemererwe. 2013. *Computational morphology and Bantu language learning: an implementation for Runyakitara*. Ph.D. thesis, University of Groningen.

Fridah Katushemererwe and Thomas Hanneforth. 2010. Fsm2 and the morphological analysis of Bantu nouns – first experiences from Runyakitara. *International Journal of Computing and ICT research*, 4(1):58–69.

Fridah Katushemererwe, Oswald K. Ndoleriire, and Shirley Byakutaaga. 2020. Morphology: General description and nominal morphology in runyakitara. In Oswald K. Ndoleriire, editor, *Runyakitara Language Studies: A Guide for Advanced Learners and Teachers of Runyakitara*, pages 33–74. Makerere University Press, Kampala, Uganda.

Ken Loge. 2015. English open word list (EOWL). `https://diginoodles.com/projects/eowl`. Accessed: 2021-02-27.

Jouni Filip Maho. 2009. NUGL Online: The online version of the New Updated Guthrie List, a referential classification of Bantu languages. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.603.6490`.

Achille Emile Meeussen. 1967. Bantu grammatical reconstructions. *Africana Linguistica*, 3(1):79–121.

George A. Miller. 1995. Wordnet: A lexical database for English. *Commun. ACM*, 38(11):39–41.

H. F. Morris and Brian Edmond Renshaw Kirwan. 1972. *A Runyankore grammar, by H. F. Morris and B. E. R. Kirwan*, revised edition. East African Literature Bureau Nairobi.

Yusuf. Mpairwe and G.K. Kahangi. 2013a. *Runyankore-Rukiga Dictionary*. Fountain Publishers, Kampala.

Yusuf. Mpairwe and G.K. Kahangi. 2013b. *Runyankore-Rukiga Grammar*. Fountain Publishers, Kampla.

Yoweri Museveni, Manuel J.K Muranga, Alice Muhoozi, Aaron Mushengyezi, and Gilbert Gomushabe. 2009. *kavunuuzi y'orunyankore/Rukiga omu Rugyeresa : Runyankore/Rukiga-English Dictionary*. Institute of Languages, Makerere University, Kampala, Uganda.

Yoweri Kaguta Museveni, Manuel Muranga, Gilbert Gumoshabe, and Alice N. K. Muhoozi. 2012. *Katondoozi y'Orunyankore-Rukiga Thesaurus of Runyankore-Rukiga*. Fountain Publishers, Kampala, Uganda.

Henry R T Muzale. 1998. *A Reconstruction of the Proto-Rutara Tense / Aspect System*. Ph.D. thesis, Memorial University of Newfoundland, Canada.

Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 216–225, Uppsala, Sweden. Association for Computational Linguistics.

Oswald K. Ndoleriire, editor. 2020. *Runyakitara Language Studies: A Guide for Advanced Learners and Teachers of Runyakitara*. Makerere University Press, Kampala, Uganda.

Tommaso Petrolito and Francis Bond. 2014. A survey of WordNet annotated corpora. In *Proceedings of the Seventh Global Wordnet Conference*, pages 236–245, Tartu, Estonia. University of Tartu Press.

Aarne Ranta. 2009a. GF: A multilingual grammar formalism. *Linguistics and Language Compass*, 3(5):1242–1265.

Aarne Ranta. 2009b. The GF Resource Grammar Library. *Linguistic Issues in Language Technology*, 2(1).

Antonio Sanfilippo. 1994. *LKB Encoding of Lexical Knowledge*, page 190–222. Cambridge University Press, USA.

Paul Schachter and Timothy Shopen. 2007. Parts-of-speech systems. In TimothyEditor Shopen, editor, *Language Typology and Syntactic Description*, 2 edition, volume 1, page 1–60. Cambridge University Press.

Gary F. Simons and D. Fennig, Charles. 2018. *Ethnologue: Languages of the world*, Twenty-first edition. SIL International, Dallas, Texas. Online version: `http://www.ethnologue.com`.

Keith Snider and James Roberts. 2004. SIL Comparative African word list (SILCAWL). *The Journal of West African Languages*, 31(2):73–122.

Ann Taylor, Mitchell Marcus, and Beatrice Santorini. 2003. The Penn treebank: An overview. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 5–22. Springer Netherlands, Dordrecht.

Charles Taylor and Yusuf Mapirwe. 2009. *A simplified Runyankore-Rukiga-English English Dictionary*. Fountain Publishers, Kampala, Uganda.

Charles V. Taylor. 1959. *A simplified Runyankore-Rukiga-English and English-Runyankore-Rukiga dictionary : in the 1955 revised orthography with tone-markings and full entries under prefixes*. Kampala : Eagle Press.

Justus Turyamwomwe. 2011. Tense and aspect in Runyankore-Rukiga, linguistic resources and analysis. Master's thesis, NTNU – Norwegian University of Science and Technology.

Richard Z. Xiao. 2008. Well-known and influential corpora. In Anke Ludeling and Merja Kyto, editors, *Corpus Linguistics: An International Handbook*, volume 1 of *Handbooks of Linguistics and Communication Science*. Mouton de Gruyter. This manuscript is not "beautified" so as to fit the publisher's stylesheet. A PDF offprint will be provided when available.

# A Appendix

```
% YAML 1.2
---
$schema: "http://json-schema.org/draft-07/schema#"
name: YAML Schema for Ry/Rk-Lex
type : seq
sequence:
  - type: map
    mapping:
      lemma:
        type: str
        required: true
      lemma_id:
        type: int
        required: true
      eng_defn:
        type: seq
        sequence:
          - type: str
        required: true
      pos:
        type: map
        mapping:
          first_level:
            type: str
            required: true
            enum:
              - verb
              - noun
              - adjective
              - adverb
              - preposition
              - pronoun
          second_level:
            type: str
            required: true
        required: true
      synonyms:
        type: seq
        required: false
        sequence:
          - type: str
      lang:
        type: str
        required: true
        enum:
          - all
          - nyn
          - cgg
      conjugations:
        type: seq
        sequence:
          - type: map
            mapping:
              nyn:
                type: str
                required: false
              cgg:
                type: str
                required: false
              all:
                type: str
                required: false
        required: false
      noun_classes:
        type: seq
        sequence:
          - type: str
        required: false
```

# A Supervised Machine Learning Approach for Post-OCR Error Detection for Historical Text

**Dana Dannélls**
Språkbanken Text
University of Gothenburg
dana.dannells@svenska.gu.se

**Shafqat Mumtaz Virk**
Språkbanken Text
University of Gothenburg
shafqat.virk@svenska.gu.se

## Abstract

Training machine learning models with high accuracy requires careful feature engineering, which involves finding the best feature combinations and extracting their values from the data. The task becomes extremely laborious for specific problems such as post Optical Character Recognition (OCR) error detection because of the diversity of errors in the data. In this paper we present a machine learning approach which exploits character n-gram statistics as the only feature for the OCR error detection task. Our method achieves a significant improvement over the baseline reaching state-of-the-art results of 91% and 89% F1 score on English and Swedish datasets respectively. We report various experiments to select the appropriate machine learning algorithm and to compare our approach to previously reported traditional approaches.

## 1 Introduction

Post processing is a conventional approach for correcting errors that are caused by Optical Character Recognition (OCR) systems. Traditionally, the task is divided into two subtasks: (1) Error detection, classify words as either erroneous or valid, and (2) Error correction, find suitable candidates to correct the erroneous words (Kolak and Resnik, 2005; Kissos and Dershowitz, 2016; Mei et al., 2016). Previous research has shown that machine learning based approaches are suitable for both subtasks (Schulz and Kuhn, 2017; Nguyen et al., 2018, 2019a; Dannélls and Persson, 2020). In the current work we aim to improve on the first task for historical texts by using machine learning techniques.

Training an accurate machine learning model requires handcrafted feature engineering,[1] which involves finding the best feature combinations and parameter settings. In the context of post-OCR error detection, finding a suitable set of features is challenging because of the diversity of OCR errors (Amrhein and Clematide, 2018). At the same time, it is well-known that feature computation is often time and labour expensive. This raises the question: Do we always need a rich feature set for achieving better results or, depending on the task at hand, fewer features could lead to better or equally good results? To our knowledge, this question has not been addressed before.

Unlike OCR errors for modern material, the error rates for historical texts are very high, resulting from a large amount of unseen characters in the output text. This has been observed for several languages (Springmann et al., 2014; Drobac et al., 2017; Adesam et al., 2019). To address the challenges for post-OCR error detection for historical text, a number of feature combinations have previously been explored with varying success rates (more details in Section 2). In this paper, we take a different approach, and instead of trying to find the optimal set of features for the task at hand, we experimented with one n-gram character feature (Sections 3 and 4). Our method achieves a significant improvement over the baseline reaching state-of-the-art results of 91% and 89% F1 on English and Swedish datasets respectively. In addition to being simple, our approach is less expensive for feature value computations. Finally, we discuss the strengths of the method and provide pointers to future work (Section 5).

## 2 Related work

There are two approaches to OCR detection and correction. One approach incorporates fine-tuned methods for improving the OCR system. For example, Tesseract (Smith, 2007) has built-in post-

[1] The *handcrafted* part of this process is finding the most suitable features and feature combinations by examining the data manually.

|  | Character | Word n-gram | Context | Features tot. | Method | Recall (%) |
|---|---|---|---|---|---|---|
| Mei et al. (2016) | ✓ | ✓ | ✓ | 6 | RM | 73.9 |
| Khirbat (2017) | ✓ | ✓ | ✓ | 3 | SVM | 44.2 |
| Nguyen et al. (2019b) | ✓ | ✓ | ✓ | 13 | GTB | 61 & 76 |
| Dannélls and Persson (2020) | ✓ | ✓ |  | 6 | SVM | 67 |

Table 1: Feature combinations reported in previous work on post-OCR detection using machine learning models and the percentage of detected OCR errors reported by each author (RM = Regression Model, SVM = Support Vector Machine, GTB = Gradient Tree Boosting).

correction functions for improving the OCR results for different languages. Another approach, that is taken here and has been adapted by the majority of previous works, builds on the output results of a specific OCR system – the one being referred to as post-OCR processing. The obvious advantage of the latter approach is that the developed method is not tailored to a particular system and could be applied to any OCR output regardless of the OCR system. One must bear in mind, however, that post-OCR processing is a complicated task because of the nature of the different errors produced by various OCR systems.

The majority of post-OCR methods of error detection exploits supervised (Evershed and Fitch, 2014; Drobac et al., 2017; Khirbat, 2017) or unsupervised (Hammarström et al., 2017; Duong et al., 2020) machine learning techniques, depending on whether the ground truth data is available or not. In this paper we focus on supervised methods. The methods described below have been trained on each word of the document. Words have been classified as either erroneous or correct. Precision, recall and F-score have been calculated based on the predicted erroneous words.

Mei et al. (2016) have experimented with 6 features containing character, word n-gram and context information. They have reported a recall for bounded (true punctuation) detection of 73.5% using regression models. Khirbat (2017) has trained a support vector machine (SVM) model with 3 features: presence of non alpha-numeric characters, bi-gram frequencies of the word and context information, that is if the word appears with its context in other places. He reported 69.6% precision, 44.2% recall and 54.1% F1. Nguyen et al. (2019b) experimented with 13 character and word features on two datasets of handwritten historical English documents (monograph and periodical) taken from the ICDAR competition (Chiron et al., 2017). The features they have experimented with include character and word n-gram frequencies, part-of-speech, and the frequency of the OCR token in its candidate generation sets which they generated using edit-distance and regression model. They trained a Gradient Tree Boosting classifier and achieved a recall of 61% and 76% and an F1 of 70% and 79% on each dataset respectively. Their results are the highest reported on the ICDAR English dataset.

Dannélls and Persson (2020) have trained an SVM model and experimented with 6 statistical and word based features, including the number of non-alphanumeric characters, number of vowels, word length, tri-gram character frequencies, number of uppercase characters and the amount of numbers occurring in the word. They reported 67% recall, and 63% F1, which is the highest results reported on Swedish text from the 19th century.

An overview of the feature sets previous authors have experimented with and the recall of the error detection machine learning models reported by each is provided in Table 1.

## 3 Method

### 3.1 Datasets

We experimented with three datasets, two for English and one for Swedish.

The first English dataset (henceforth Sydney) comprises newspaper text from the Sydney Morning Herald 1842-1954, consisting of 10,498,979 tokens and a ground truth data of randomly sampled paragraphs (Evershed and Fitch, 2014). The material was processed with Abbyy Finereader 14. The training and testing sets compiled from this material contain instances from this particular OCR system only.

The second English dataset (henceforth ICDAR2017) is the monograph dataset from the ICDAR 2017 competition (Chiron et al., 2017), which accounts for 754,025 OCRed tokens with their corresponding ground truth.[2] The dataset has been collected from national libraries and university collections. It was processed with Abbyy Finereader 11, and the ground truth comes from various European project initiatives.

The Swedish dataset (henceforth Fraktur&Olof) consists of a selection of digitized versions of older Fraktur prints from 1626-1816,[3] and all pages from Olof v. Dalin's Swänska Argus from 1732-1734,[4] all amounting to 261,323 tokens. The ground truth for this dataset was produced through double-keying. The material was processed with three OCR systems: Abbyy Finereader 12, Tesseract 4.0 and Ocropus 1.3.3. Each one of these systems is using their own built-in dictionary and the quality of the OCR results differs significantly between the systems. When we compiled the training and testing sets in our experiments, described in Section 4, we included instances from all three systems to avoid the risk of developing a method that is biased towards a particular OCR system (Dannélls and Persson, 2020).[5]

In our experiments (see Section 4), we chose randomly selected subsets of 50K tokens from the Sydney and the Fraktur&Olof datasets. A balanced set of 92K instances was selected from the ICDAR2017 dataset. All three subsets were then divided into training (80%) and test (20%) sets. Depending on the vocabulary size, it can take days to run the models. Because of this constraint the complete datasets were not used in the experiments.

### 3.2 Preprocessing

All of the above datasets come in different formats, therefore we had to preprocess them before we could proceed. For our experiments we needed to first align the OCRed and ground truth data at the token level and secondly convert the aligned data to feature vectors.

In the ICDAR2017 and Sydney datasets, the OCRed and ground truth data are aligned at the character level. To align them at the token level, the ground truth was tokenized on space, and for each token the same number of characters was extracted from the OCRed version. After removing the special alignment symbols ('@' and '#') that were inserted by the competition organizers, the resulting OCRed and ground truth tokens were compared to set the labels: '0' if the token was erroneous or '1' if the token was valid.[6] These labels are to be learned and predicted by the machine learning models during training and testing. Learning is based on a set of feature combinations to help the model detect the errors in the output of the OCR, described in Section 4.

The tokens in the Swedish dataset were computed by first removing duplicate white-spaces and second, replacing all non-space white-spaces such as tab with space. Then, valid tokens were extracted from the ground truth data and were assigned label '1'. Erroneous tokens were extracted from the OCRed data and were compared to a large scale computational Swedish lexicon (Borin and Forsberg, 2011). If the token appeared in the lexicon it was assigned label '1' otherwise '0'.[7]

Table 2 shows a few instances from the data produced after the preprocesscing step both for Swedish and English. The resulting full data-sets were then used to compute various features and train/test models as explained in Section 4.

| English | | | Swedish | | |
|---|---|---|---|---|---|
| Token | GT | Label | Token | GT | Label |
| matter | matter | 1 | nytta | nytta | 1 |
| the | the | 1 | sassvanter | - | 0 |
| king@ | king | 0 | angenämt | angenämt | 1 |
| very | very | 1 | p-å | - | 0 |
| glad | glad | 1 | föreställa | föreställa | 1 |
| hereof,@ | hereof, | 0 | behöfwesr | - | 0 |
| @Hkewise | likewise | 0 | Lärdomar | lärdomar | 1 |

Table 2: A sample from the English and Swedish datasets after the preprocessing step (GT = Ground Truth).

All the machine learning models we experimented with are part of the Sci-kit Python library (Pedregosa et al., 2011). Input data to all the algorithms in the sklearn library should be in numerical form, but only some of the features we experimented with are numeric (e.g. the token frequencies), the others are non-numeric (e.g. bigrams). For the non-numeric features, we used one-hot encoding for data transformation. While the details are beyond the scope of this paper, the

[2] https://drive.google.com/file/d/1-pNT00vvIqh0ss_5b2aHo-nG8advaFJi/view

[3] https://spraakbanken.gu.se/en/resources/svensk-fraktur-1626-1816

[4] https://spraakbanken.gu.se/en/resources/dalin-then-swaanska-argus-1732-1734

[5] Datasets are available under CC-BY license and can be accessed from https://spraakbanken.gu.se/en/resources#refdata.

[6] Valid OCRed tokens are identical to the GT token.

[7] Because preprocessing of the datasets is completely automatic, we noticed that a small proportion of instances was miss-classified.

| | Fraktur&Olof | | | Sydney | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 |
| Logistic Regression | 0.82 | 0.74 | 0.76 | 0.74 | 0.60 | 0.65 |
| Decision Tree | 0.84 | 0.79 | **0.80** | 0.71 | 0.73 | **0.71** |
| Bernoulli Naive Bayes | 0.84 | 0.78 | 0.79 | 0.79 | 0.58 | 0.63 |
| Naive Bayes | 0.67 | 0.54 | 0.37 | 0.70 | 0.60 | 0.59 |
| SVM | 0.84 | 0.79 | **0.80** | 0.74 | 0.60 | 0.66 |

Table 3: Evaluation results of error detection for English and Swedish datasets trained with different models on one feature. The best performing models are highlighted in bold (Experiment I).

major idea behind one-hot encoding is to add an extra dimension in the feature vector for each unique feature value. This produces an N dimensional feature vector (the learned encoding), where N is the total number of unique values of the complete feature set. An instance is then encoded by setting the dimension corresponding to the feature value to '1', while the remaining dimensions are set to '0'. We used sklearn's 'CountVectorizer' and 'SVC' classifiers with default parameter to learn the encoding and train the different machine learning models. In all the experiments we used the default SVM *radial basis* kernel function.

## 4 Experiments and results

We devised three experimental settings. The first experiment is set up to learn which machine learning algorithm performs best on the OCR error detection task. In the second experiment we create our baseline and train a machine learning model with different feature configurations. Given our findings in the second experiment we further explore the best performing configuration with simple character n-gram features.

### 4.1 Experiment Setup

**Experiment I** Machine learning classifiers are known to have pros and cons depending on the task. To our knowledge, there are no previous studies to examine the performance of different machine learning techniques for detecting OCR errors. We compared between 5 popular state-of-the-art machine learning classifiers to learn which of them is most suitable for this task. More specifically, we explored Logistic Regression, Decision Tree, Bernoulli Naive Bayes, Naive Bayes and SVM.

Logistic Regression has been very common for binary tasks because of its success in linearly separating data. Decision Tree is a predictive classifier, most widely used for solving inductive problems. It has also proven to be efficient for detecting OCR errors (Abuhaiba, 2006). Both Bernoulli Naive Bayes and Naive Bayes are probabilistic classifiers. Bernoulli Naive Bayes includes a probability for whether a term is in the data or not, and therefore has been shown useful for document classification. SVM is a supervised machine learning method that is very effective in high dimensional spaces. It has gained high popularity for detecting OCR errors partially because its performance has proven to be as robust and accurate as of a neural network (Arora et al., 2010; Hamid and Sjarif, 2017; Amrhein and Clematide, 2018).

In this experimental setting, we trained all machine leaning classifiers on one feature that is the actual word. For training and testing, 5-cross validation was applied. Because of the time needed to train the models, the classifiers were only trained on two datasets, Fraktur&Olof and Sydney.

**Experiment II** We experimented in three different settings. First, we form our baseline by training the best performing model (from experiment I) on the 6 features reported by Dannélls and Persson (2020). This set of features forms our baseline, it includes: (1) whether the word contains an alphanumeric character, (2) the word tri-gram frequency, (3) whether the word contains a vowel, (4) whether the word length is over 13 characters (5) whether the first letter appears in upper case, (6) whether the word contains a number. Since all of the features are numeric in nature, no encoding was required for this setting.

Second, analogous to previous approaches (Mei et al., 2016; Khirbat, 2017; Nguyen et al., 2019b), we enhanced the feature set with 4 additional features (referred to as the 10-feature model): (1) the actual word (2) the actual word length, (3) context, i.e. the word proceeding and following the actual word, (4) whether the word appears in the word2vec model, here we apply a simple look-up

| | Fraktur&Olof | | | Sydney | | | ICDAR2017 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 |
| Baseline | 0.82 | 0.68 | 0.70 | 0.73 | 0.59 | 0.60 | 0.85 | 0.85 | 0.85 |
| 10-feature | 0.80 | 0.71 | 0.73 | 0.81 | 0.62 | 0.63 | NA | NA | NA |
| 1-word-feature | 0.78 | 0.83 | 0.79 | 0.80 | 0.62 | 0.63 | 0.86 | 0.84 | 0.84 |

Table 4: Evaluation results of error detection with SVM, once computed with the 10-feature model and once with the 1-word-feature model (Experiment II). Baseline was computed with the 6-feature model.

method against the pre-trained model by Hengchen et al. (2019). In this case, some of the features (e.g. the word itself) are non-numeric, hence one-hot encoding was applied for those features. As mentioned previously, this means adding an extra dimension for each unique word in the training data to learn the encoding and then encoding each instance by setting the corresponding dimension values accordingly. The same applies for the context feature.

Third, we removed all features and trained the model only on one feature, the actual word (referred to as the 1-word-feature model).[8] This potentially means turning the model into a dictionary look-up kind of system, with the major restriction that the system is not scalable and is restricted to only those words which have been seen in the training data.

**Experiment III** To overcome the above mentioned limitation of using the word as the only feature, we experimented further with n-gram feature sets. For each candidate word, we generated character uni-, bi-, and tri-grams first, and then their counts within the word were used as feature values to train the model. To take an example, suppose our candidate word is ‘passenger’, the computed uni-, bi-, and tri-gram features vectors will be as follows:

- **uni-gram** {‘a’:1, ‘e’:2, ‘g’:1, ‘n’:1, ‘p’:1, ‘r’:1, ‘s’:2}
- **bi-gram** {‘p’:1, ‘as’:1, ‘en’:1, ‘er’:1, ‘ge’:1, ‘ng’:1, ‘pa’:1, ‘r ’:1, ‘se’:1, ‘ss’:1}
- **tri-gram** {‘pa’:1, ‘ass’:1, ‘eng’:1, ‘er ’:1, ‘ger’:1, ‘nge’:1, ‘pas’:1, ‘sen’:1, ‘sse’:1}

The intuition is simple: It is more probable that the corresponding uni-, bi-, and tri-grams have been seen in the training data as opposed to the complete word. This can remove the above described limitation and make the system more scalable. The models were then trained on the resulting feature vectors and then tested on the test data.

### 4.2 Results

**Experiment I** The results from the first experiment, where only one feature was used to train different machine learning models, are presented in Table 3. We can observe that both Decision Tree and SVM outperform the other models on the Swedish dataset, achieving 80% F1. Bernoulli Naive Bayes is almost as good with an F1 of 79%. Decision Tree is the best performing model on the English dataset with the highest F1 of 71%. These results strengthen previous successful attempts to train an SVM model for detecting OCR errors (Arora et al., 2010; Hamid and Sjarif, 2017; Clematide and Ströbel, 2018).

**Experiment II** The results from the second experiment are presented in Table 4. Even though we experimented with the same feature combination as reported in Dannélls and Persson (2020), our baseline yields 70% F1 compared to their reported 63% F1 probably owing to parameter settings and the chosen sub datasets. The results on Fraktur&Olof show that the model trained on 1-word-feature outperforms the model trained on 6 (baseline) and 10 feature sets respectively.

Interestingly, the results on the Sydney dataset show no difference in performance between the 10-feature and the 1-word-feature datasets. In contrast to the Fraktur&Olof dataset where F1 increases with 5%. We believe the difference in the results between Fraktur&Olof and Sydney can be characterized by the nature of the data. A manual inspection of the datasets reveals that Fraktur&Olof is representative with regards to its vocabulary. Hence, more words in the Swedish dataset were seen in the training set as compared to the English counterpart.

Our baseline results on the ICDAR2017 dataset

[8] We write ‘word’ although, in practice, it actually refers to a token because ‘a word’ is not necessarily a lexical word, for example if we consider an instance from our training data, i.e. ‘ycsteidas’.

|  | Fraktur&Olof |  |  | Sydney |  |  | ICDAR2017 |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 |
| Uni-gram | 0.81 | 0.78 | 0.78 | 0.83 | 0.68 | 0.70 | 0.89 | 0.87 | 0.87 |
| Bi-gram | 0.87 | 0.87 | 0.87 | 0.86 | 0.76 | **0.79** | 0.91 | 0.91 | **0.91** |
| Tri-gram | 0.89 | 0.89 | **0.89** | 0.84 | 0.74 | 0.77 | 0.88 | 0.88 | 0.88 |

Table 5: The accuracy scores of the SVM classifier trained with the n-gram feature sets. Best results for each dataset in bold (Experiment III).

are not as high compared to the F1 reported by Mei et al. (2016) and Nguyen et al. (2019b). The reason for this is because we are experimenting with completely different datasets with respect to both size and content. Training the SVM classifier on 1-word-feature did not improve the baseline. This, again, may be due to the nature of the data.

**Experiment III** The results from the experiments with the n-gram feature sets are shown in Table 5. When we compare between the results of the 1-word-feature and the n-gram feature models, we see there is an improvement for all three datasets: Fraktur&Olof, Sydney, and ICDAR2017.

The best performance achieved on Fraktur&Olof is 89% F1 with the tri-gram model. This is the highest results on 19th century Swedish text reported so far. The best performing model for Sydney is 79% F1, achieved with the bi-gram model. The best results achieved on the ICDAR2017 data are also with the bi-gram model. For all datasets the n-gram models show an incremental improvement. One explanation for the difference between the results might be the differences between the types of OCR errors in each dataset. The most obvious errors on Fraktur&Olof are due to appearance of long *s*, uppercase letters and miss-recognition of the Swedish vowels ('å' and 'ä'), while obvious errors in ICDAR2017 are due to hypens and non-alphanumeric characters.

## 5 Discussion and Conclusion

Training supervised machine learning models with large number of features is a computationally expensive task. This has been demonstrated in previous work where carefully crafted features were considered at the expense of high computational costs. In our experiments we trained an SVM model on a number of feature sets consisting of 6 features, 10 features, one word feature and three n-gram character level features, and compared their results. By training the model on the word itself, we are necessarily turning the machine learning model into a dictionary look-up kind of system. The results show that the 1-word-feature model trained on word level is sufficient, not only for improving over the baseline, but also for reaching better results than previously reported for historical Swedish data. The results on the English datasets show that the 1-word-feature model is as good as the 10-feature model. This proves that with the dictionary of words over the training data alone we can better predict whether a word contains an OCR error or not. However, this type of approach has its own limitations as mentioned previously, and for that purpose, we turned to character level n-gram based approach, which improved the results further.

What makes the proposed approach interesting is that it eliminates the need to compute many features for detecting OCR errors. On the other hand, we are aware that it relies on the availability of large amount of training data which is also costly, and will in turn also increase the training time.

Notwithstanding, in this work we kept the datasets rather small mostly because of time constraints and memory issues. This leaves several open questions regarding the representativeness of the chosen data. Correspondingly, we are unable to make direct comparisons with the results reported by others. In the future, we plan to experiment with bigger datasets, and our hope is to improve on the results reported in this study. Parameter optimization of the chosen machine learning algorithms is another direction which can be explored further to improve the results in future. Another possible way to improve the results is to use the back-off approach in the n-gram setting. Taking a back-off approach we will use a bi-gram if a tri-gram is not in the vocabulary in a tri-gram setting, and likewise a uni-gram if a bi-gram is not in the vocabulary.

## Acknowledgments

The work presented here was funded by (1) the *Dictionary/Grammar Reading Machine: Compu-*

*tational Tools for Accessing the World's Linguistic Heritage* (DReaM) Project awarded 2018–2020 by the Joint Programming Initiative in Cultural Heritage and Global Change, Digital Heritage and Riksantikvarieämbetet, Sweden; (2) the Swedish Research Council as a part of the project *South Asia as a linguistic area? Exploring big-data methods in areal and genetic linguistics* (2015–2019, contract no. 421-2014-969); (3) *From Dust to Dawn: Multilingual Grammar Extraction from Grammars* project funded by Stiftelsen Marcus och Amalia Wallenbergs Minnesfond 2007.0105, Uppsala University; (4) the Swedish Research Council as part of the project *Evaluation and refinement of an enhanced OCR-process for mass digitisation* (2019–2020, grant agreements IN18-0940:1 and 421-2014-969). It is also supported by Språkbanken Text and Swe-Clarin, a Swedish consortium in Common Language Resources and Technology Infrastructure (CLARIN) Swedish CLARIN (grant agreement 821-2013-2003). The authors would like to thank the SLTC anonymous reviewers for their valuable comments and suggestions on how to improve the paper.

## References

Ibrahim S I Abuhaiba. 2006. Efficient OCR using simple features and decision trees with backtracking. *Journal for science and engineering*, 31(2):223–244.

Yvonne Adesam, Dana Dannélls, and Nina Tahmasebi. 2019. Exploring the quality of the digital historical newspaper archive kubhist. In *Proceedings of the 4th Conference of The Association Digital Humanities in the Nordic Countries (DHN)*, pages 9–17, Copenhagen, Denmark. University of Copenhagen, Faculty of Humanities.

Chantal Amrhein and Simon Clematide. 2018. Supervised OCR error detection and correction using statistical and neural machine translation methods. *Journal for Language Technology and Computational Linguistics (JLCL)*, 33(1):49–76.

Sandhya Arora, Debotosh Bhattacharjee, Mita Nasipuri, Latesh L. G. Malik, Kundu Mahantapas, and Dipak Kumar Basu. 2010. Performance comparison of SVM and ANN for handwritten Devnagari character recognition. *IJCSI International Journal of Computer Science Issues*, 7(6).

Lars Borin and Markus Forsberg. 2011. A diachronic computational lexical resource for 800 years of Swedish. In Caroline Sporleder, Antal van den Bosch, and Kalliopi Zervanou, editors, *Language technology for cultural heritage*, pages 41–61. Springer, Berlin.

Guillaume Chiron, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. 2017. ICDAR2017 competition on post-OCR text correction. *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01:1423–1428.

Simon Clematide and Phillip Ströbel. 2018. Improving OCR quality of historical newspapers with handwritten text recognition models. In *Workshop DARIAH-CH*, Neuchâtel. University of Zurich.

Dana Dannélls and Simon Persson. 2020. Supervised OCR post-correction of historical Swedish texts: What role does the OCR system play? In *Proceedings of the Digital Humanities in the Nordic Countries 5th Conference*, volume 2612 of *CEUR Workshop Proceedings*, pages 24–37, Riga, Latvia. CEUR-WS.org.

Senka Drobac, Pekka Kauppinen, and Krister Lindén. 2017. OCR and post-correction of historical Finnish texts. In *Proceedings of the 21st Nordic Conference on Computational Linguistics (Nodalida)*, pages 70–76, Gothenburg, Sweden. Association for Computational Linguistics.

Quan Duong, Mika Hämäläinen, and Simon Hengchen. 2020. An unsupervised method for OCR post-correction and spelling normalisation for Finnish. *arXiv*, abs/2011.03502.

John Evershed and Kent Fitch. 2014. Correcting noisy OCR: Context beats confusion. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, DATeCH '14, pages 45–51, New York, NY, USA. Association for Computing Machinery.

Norhidayu Abdul Hamid and Nilam Nur Amir Sjarif. 2017. Handwritten recognition using SVM, KNN and Neural Network. *ArXiv pre-print*, abs/1702.00723.

Harald Hammarström, Shafqat Mumtaz Virk, and Markus Forsberg. 2017. Poor man's OCR post-correction: Unsupervised recognition of variant spelling applied to a multilingual document collection. In *Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage*, DATeCH 2017, pages 71–75, NY, USA.

Simon Hengchen, Ruben Ros, and Jani Marjanen. 2019. A data-driven approach to the changing vocabulary of the 'nation' in English, Dutch, Swedish and Finnish newspapers, 1750-1950. In *Proceedings of the Digital Humanities (DH) conference 2019, Utrecht, The Netherlands.*

Gitansh Khirbat. 2017. OCR post-processing text correction using simulated annealing (OPTeCA). In *Proceedings of the Australasian Language Technology Association Workshop 2017*, pages 119–123. Association for Computational Linguistics.

Ido Kissos and Nachum Dershowitz. 2016. OCR error correction using character correction and feature-based word classification. In *Document Analysis Systems 12th IAPR Workshop*, pages 198–203. IEEE.

Okan Kolak and Philip Resnik. 2005. OCR post-processing for low density languages. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 867–874, Vancouver, B.C., Canada. Association for Computational Linguistics.

Jie Mei, Aminul Islam, Yajing Wu, Abidalrahman Mohd, and Evangelos E Milios. 2016. Statistical learning for OCR text correction. *arXiv preprint*, abs/1611.06950.

Thi-Tuyet-Hai Nguyen, Mickaël Coustaty, Doucet Antoine, and Nhu-Van Nguyen. 2018. Adaptive edit-distance and regression approach for post-OCR text correction. In *20th International Conference on Asia-Pacific Digital Libraries, ICADL*, volume 11279 of *Lecture Notes in Computer Science*, pages 278–289. Springer.

Thi-Tuyet-Hai Nguyen, Adam Jatowt, Mickael Coustaty, Nhu-Van Nguyen, and Antoine Doucet. 2019a. Deep statistical analysis of OCR errors for effective post-OCR processing. In *Proceedings of the 18th Joint Conference on Digital Libraries*, JCDL '19, page 29–38. IEEE Press.

Thi-Tuyet-Hai Nguyen, Adam Jatowt, Mickaël Coustaty, Nhu-Van Nguyen, and Antoine Doucet. 2019b. Post-OCR error detection by generating plausible candidates. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 876–881. IEEE.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12:2825–2830.

Sarah Schulz and Jonas Kuhn. 2017. Multi-modular domain-tailored OCR post-correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2716–2726, Copenhagen, Denmark. Association for Computational Linguistics.

Ray Smith. 2007. An overview of the Tesseract OCR engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE.

Ume Springmann, Dietmar Najock, Hermann Morgenroth, Helmut Schmid, Annette Gotscharek, and Florian Fink. 2014. OCR of historical printings of Latin texts: Problems, prospects, progress. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, DATeCH 2014, pages 71–75, New York, USA. Association for Computing Machinery.

# Automated Writing Support for Swedish Learners

**Yaroslav Getman**
Department of Signal Processing and Acoustics
Aalto University
yaroslav.getman@aalto.fi

## Abstract

This paper describes a tool developed for lexical and grammatical analysis of Swedish text and providing automated feedback for language learners. The system looks for words and word sequences that are likely to contain errors and suggests how to correct them using different non-neural models. The feedback consists of alternative word and word sequence suggestions and morphological features[1] which need to be corrected. Although the system is able to provide reasonable feedback which is believed to be useful for language learners, it still needs further improvements to address the drawbacks such as low precision.

## 1 Introduction

The majority of automatic error detection and correction systems focus on searching for mistakes and providing right solutions directly without any feedback. Instead, providing the feedback would be useful especially for non-native writers and help them to understand the mistakes and correct the errors on their own.

In the *DigiTala* project (2019–2023), financed by Academy of Finland, we are developing tools for automatic Finnish and Swedish spoken language proficiency evaluation of non-native speakers. This paper addresses a system built for lexical and grammatical analysis of Swedish and giving automatic supportive feedback for language learners.

For the analysis, the current version of the system involves non-neural models only, supposing that they are able to provide enough accuracy while requiring less training data than deep neural networks. However, the models can be replaced later by neural ones for future experiments.

The rest of the paper is organized as follows. Section 2 provides a brief overview of related research. Section 3 describes the system components and the error analysis. Section 4 presents an example analysis performed by the system. Section 5 concludes the paper with ideas for future work.

## 2 Related work

There are some systems which act as pedagogical tools and provide constructive feedback, such as one developed by Morgado da Costa et al. (2020) for assisting students in their scientific English writing. The system described in the paper uses computational parsers and general NLP techniques, e.g. checking for repeated words, sentence length, word capitalization, etc.

Several grammar checkers for second language writers of Swedish have been developed in the research project *CrossCheck* (Bigert et al., 2004). One of them, called *Granska* (Domeij et al., 2000), consists of a POS tagger, a spelling checker and manually constructed rules for error detection and correction. The second one, *ProbGranska* (Bigert and Knutsson, 2002), is a statistical method which searches for unlikely grammatical constructions using POS tag trigram frequencies. The third one, *SnålGranska* (Sjöbergh, 2005), is a weakly supervised machine learning based system trained on a text corpus with artificially created errors.

A system called *Revita* (Katinskaia et al., 2017, 2018) is designed to support language learning and focuses primarily on Finno-Ugric languages. The system automatically generates ”cloze” exercises from texts, where a language learner needs to fill in the missing words to the sentences.

For German language learners, there is a feedback mechanism developed by Rudzewitz et al. (2018) as a part of language learners’ tutoring system *FeedBook* (Rudzewitz et al., 2017). *FeedBook*

[1] https://universaldependencies.org/treebanks/sv_talbanken/index.html#features

consists of short answer and fill-in-the-blanks exercises. The system compares student answers to the target answers stored for each task and generates feedback based on predefined error templates for five grammar error types: tenses, comparatives, gerunds, relative clauses, reflexive pronouns.

## 3 System description

The tool developed so far relies on a language model (LM), when looking for errors in the input sentences. While ngrams (contiguous sequences of *n* words) which are present in the LM are supposed to be correct, unknown ngrams and out-of-vocabulary (OOV) words can possibly contain errors. If OOV words or unknown bigrams (two-word sequences) are found from the sentences, they are examined by the system in more detail and the feedback is provided. For the OOV words found in the sentences, the tool proposes similar words. Also, it suggests most likely part of speech and morphological features, or, grammatical categories (grammatical case, person, number, etc.), to use when asking to replace the OOV word with another word. If unknown bigram is detected, the system searches for similar bigrams and asks to change the part of speech and/or correct morphological features, if needed.

### 3.1 Corpora and Models

In total, 6 models are used in this work for different purposes: a part-of-speech (POS) tagger, a morphological features tagging module, a word-level LM, a subword-level LM, a LM trained on POS tags and a model for word segmentation.

A pretrained morphological features tagging module from the Stanza library (Qi et al., 2020) was also used in this work. The module is based on the Swedish-Talbanken treebank[2]. The treebank has 6,026 sentences and 96,819 tokens. It was used also for training a Conditional Random Fields (CRF) POS tagger.

The Swedish YLE corpus[3] was used for training other models. The corpus is a collection of news articles published by Finland's national public broadcasting company in Swedish from the year 2012 to the year 2018. It consists of 6,810,509 sentences and 93,405,178 tokens. The vocabulary has 1,102,561 words. The data was converted to a lowercased plain text corpus with punctuation preserved. Keeping the punctuation in the text corpus is important, because otherwise, a lot of words would form grammatically incorrect bigrams, e.g. the last word of one sentence and the first word of the following sentence which are not necessarily related to each other. When it comes to evaluation of transcribed speech, post-processing techniques for restoring the punctuation in the transcripts can be considered in order for the system to provide more proper analysis results.

In addition to plain text, the source data of the Swedish YLE corpus contains positional attributes for each word such as number of the token within the sentence, lemma, POS tag, morphological analysis, dependency head number and dependency relation. The annotations were extracted separately to create a new corpus consisting of POS tag sequences which was then used for training a trigram POS LM.

Morfessor 2.0 (Smit et al., 2014) is a tool for unsupervised and semi-supervised statistical morphological segmentation. In this work, a Morfessor model was trained in unsupervised manner. The whole YLE corpus was then passed through the model to divide it into subwords and train a subword-level trigram LM.

### 3.2 OOV words analysis

An OOV word found from the text is first divided into segments using the Morfessor model. Depending on the word length, different number of possible segmentations is used for further analysis: 5 most likely segmentations are preserved for words consisting of 5 or more characters and *N* = *length of word* segmentations for words shorter than 5 characters. It should be taken into account that different configurations were tested and these numbers found to be optimal for Swedish and might need to be adjusted for other languages.

For each of these segmentations, a new word is formed by reducing the last segment from the OOV word. If the new word is not found from the vocabulary of the LM, it is tested for possible continuations by the subword-level trigram LM: the system searches for most likely next segment(s) based on the previous segment or two previous segments. If such segments are found, new words are formed. The tool then checks if these new words are found from the LM vocabulary.

[2] https://github.com/UniversalDependencies/UD_Swedish-Talbanken

[3] http://urn.fi/urn:nbn:fi:lb-2020021103

A trigram POS LM is used to find the most likely part of speech given two preceding POS tags. If no POS tag can follow the previous POS tags according to the POS LM model, the most likely POS tag given one preceding POS tag is used instead. Morphological features are suggested using the bigram LM and a morphological features tagging module: 20 words which are most likely to follow the word before the OOV word are collected using the LM. Then, only the words belonging to the most likely part of speech are preserved. For the these words, morphological features are extracted and the most frequent value for each feature is selected.

### 3.3 Unknown bigrams analysis

To find bigrams similar to an unknown bigram, 5 most likely word segmentations are collected using the Morfessor model for each word of the bigram. In each of these segmentations, only the longest segment consisting of at least 3 characters is preserved. If a word consists of less than 3 characters, the whole word is preserved. After that, similar words are collected by searching for words containing any of these segments from the LM vocabulary. Then, combinations of these words are formed, including the combinations of the first word of the bigram with words similar to the second word of the bigram. The LM analyzes each of these new bigrams: the ones that are possible according to the LM are then preserved and proposed by the system as bigrams similar to the initial unknown bigram. If no similar bigrams are found, the process runs from the beginning with the number of word segmentations to collect increased by one each time until at least one similar bigram is found or the manually set threshold for the maximum number of word segmentations to collect is reached.

The tool also suggests part of speech and morphological features to use when replacing the second word of the unknown bigram. It selects the POS tag and the values for the morphological features in a similar way as for OOV words. However, the part of speech of the second word of the bigram is also taken into account. The system compares the probability for the POS tag of the word to follow two preceding POS tags to the mean of the probabilities of all possible POS tags that can follow the corresponding POS tag sequence.

If the probability is above the average, the system supposes that this part of speech is likely enough and suggests to use another word belonging to the same part of speech. In this case, morphological features of the second word of the bigram are extracted and compared to the most likely morphological features. The most likely features are collected in a similar way as for OOV words. However, here all possible values are preserved instead of selecting the most common value for each feature. The most common value is selected and suggested to use only in case if the value of the feature for the word is not in the list of possible values.

If the probability is below the average, the tool suggests to use the POS tag which is most likely to follow two preceding POS tags. The most likely morphological features are then collected in the same way as for OOV words. If no POS trigrams are found from the POS LM, the system searches for POS bigrams instead.

## 4 Examples of System Output

In this section, we will use the following example to explore the output of the tool:

(1) *Hej Peter! Jag försökte ringa du, men din
Hi Peter! I tried to call you, but your
mobilen var avstängd.
phone was switched off.
*‘Hi Peter! I tried to call you, but your phone was switched off.’*

When the example is fed to the system, the output is:

”***Text to evaluate:*** *Hej Peter! Jag försökte ringa du, men din mobilen var avstängd.*

***(For human evaluator) Text tagged with part-of-speech tags:***

*[[(’Hej’, ’IN’), (’Peter’, ’PM’), (’!’, ’MAD’)], [(’Jag’, ’PN’), (’försökte’, ’VB’), (’ringa’, ’VB’), (’du’, ’PN’), (’,’, ’MID’), (’men’, ’KN’), (’din’, ’PS’), (’mobilen’, ’NN’), (’var’, ’VB’), (’avstängd’, ’PC’), (’.’, ’MAD’)]]*

***Analyzed text with annotations:***

*hej peter ! jag försökte ringa du***1** , men din mobilen***2** var avstängd .*

***Unknown words:***

***Uncommon ngrams:***

***1.*** *ringa du. Similar ngrams: placeringar du, placeringar fördubblades, beväringarna duschar, tillbringat dubbelt, tillbringade du. You used the pronoun du (Case: Nom, Definite: Def, Gender: Com, Number: Sing, PronType: Prs). You can also try to use some other pronoun instead of du. It is also recommended to correct the following morphological featutes: Case: Acc.*

***2.*** *din mobilen. Similar ngrams: din mobiltelefon, din mobil, din mobila, din mobilbutik, din mobils. You used the noun mobilen (Case: Nom, Definite: Def, Gender: Com, Number: Sing). You can also try to use some other noun instead of mobilen. It is also recommended to correct the following morphological featutes: Definite: Ind.*"

In this example, there are two grammatical errors: ***"*ringa du"*** (*"call you"*), where the pronoun ***du*** should be used in the accusative case (***dig***), and ***"*din mobilen"*** (*"your phone"*), where the noun should be in the indefinite form (***mobil***). The system found both errors and marked the bigrams containing the errors as uncommon.

As can be seen from the list of similar ngrams, the system did not manage to provide the correct bigram (***"ringa dig"***). This happens because the morphological segmentation model defines the word ***du*** itself as its only morpheme which is then used as a search query in the LM vocabulary. However, the tool suggests correctly to change the case of the pronoun from *nominative* to *accusative*. For the second incorrect bigram, the system managed to provide both correct bigram (***"din mobil"***) and suggestion to change the form of the noun from *definite* to *indefinite*.

Here is another example for the analysis:

(2) *Ursäkta mig, jag kan inte komma eftersom
Excuse me, I can not come because
jag blev sjuk. Kan vi träffas
I got sick. Can we meet
i övermorgon eller kanske i
the day after tomorrow or maybe in
fredag?
Friday?
*'Excuse me, I can not come because I got sick. Can we meet the day after tomorrow or maybe in Friday?'*

and the corresponding output of the system:

"***Text to evaluate:*** *Ursäkta mig, jag kan inte komma eftersom jag blev sjuk. Kan vi träffas i övermorgon eller kanske i fredag?*

***(For human evaluator) Text tagged with part-of-speech tags:***

*[[('Ursäkta', 'VB'), ('mig', 'PN'), (',', 'MID'), ('jag', 'PN'), ('kan', 'VB'), ('inte', 'AB'), ('komma', 'VB'), ('eftersom', 'SN'), ('jag', 'PN'), ('blev', 'VB'), ('sjuk', 'JJ'), ('.', 'MAD')], [('Kan', 'VB'), ('vi', 'PN'), ('träffas', 'VB'), ('i', 'PP'), ('övermorgon', 'NN'), ('eller', 'KN'), ('kanske', 'AB'), ('i', 'PP'), ('fredag', 'NN'), ('?', 'MAD')]]*

***Analyzed text with annotations:***

*ursäkta mig , jag kan inte komma eftersom jag blev sjuk . kan vi träffas i övermorgon eller*1 kanske i fredag ?*

***Unknown words:***

***Uncommon ngrams:***

***1.*** *övermorgon eller. Similar ngrams: imorgon eller, imorgon håller, imorgon ställer, imorgon heller. You used the conjunction eller. You can also try to use some other conjunction instead of eller.*"

The second sentence contains an error the system did not recognise at all. An incorrect preposition **i** is used before ***fretag***, while the correct answer is ***"på fretag"*** (*"on Friday"*).

There is also a bigram marked as incorrect in the second sentence of the example: ***"övermorgon eller"*** (*"the day after tomorrow or"*). The system proposed to try another conjunction instead of ***eller***, as well as to replace the whole bigram with another one. It can be noticed that there is a bigram in the list of similar ngrams very close to the original one: ***"imorgon eller"*** (*"tomorrow or"*). Although both of them are grammatically equally correct, the original one was not found from the training set of the language model and erroneously marked as incorrect.

## 5 Discussion

The current implementation of the tool is able to analyze words and sentences at grammatical and lexical level and provide reasonable feedback. In addition, the system can be applied for other languages by replacing the models. The models can be changed also to neural ones, if needed in future. However, the work is still in progress and further improvements are needed to overcome the existing drawbacks.

Many correct words and bigrams are not recognized by the system due to morphological richness of Swedish language. However, the same word in other word form(s) can be proposed as word(s) similar to the OOV word. Many compound words are also unknown to the system. On the other hand, some common types of grammatical errors might be skipped by the system. For example, while in the previous section ***"*din mobilen"*** was recognized as incorrect bigram, the system would not recognize the error in the phrase ***"*din nya mobilen"*** ("your new phone"), since both ***"din nya"*** and ***"nya mobilen"*** are correct bigrams according to the LM. Switching to trigrams and fourgrams has helped to increase the recall of the system. However, it resulted in increased number of false positives. In other words, a lot of trigrams were recognized as incorrect due to the limited size of the training text data of the LM. Larger text corpora can help to reduce the amount of unknown words and ngrams.

The text corpus which is used for training the LM might contain lexical or grammatical errors. However, it is quite unlikely that the same error would occur many times in the corpus. Therefore, one possible solution could be to set different threshold for the log probability of a bigram. The current threshold is set to minus infinity, which means that only unknown bigrams are recognized by the system as incorrect. Setting it to a very low number close to minus infinity would possibly help to filter out some erroneous bigrams.

The tool is able to detect bigrams containing lexical errors like wrong word choice, but it cannot provide the most suitable word based on the context. Instead, the system tries to find bigrams which are similar to the original one. One possible solution to address this drawback is to use more advanced LMs, for example Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019). BERT can be used in the masked language modeling task, where an inappropriate word is masked and needs to be predicted by the LM. BERT uses both left and right context of a word and therefore is believed to make more accurate word predictions compared to the ngram LMs which look only at the preceding context. There are several BERT models available which are pretrained on Swedish text corpora, for example KB-BERT (Malmsten et al., 2020).

Because the system focuses on providing feedback, it is difficult to evaluate how well it works. In addition, there is lack of labeled data for grammatical error correction task in general, and no such a dataset was found for Swedish. A corpus of Swedish learner essays where errors are manually annotated has been presented as part of the SweLL (Volodina et al., 2016) project. Unfortunately, at the time of writing, the corpus was not yet available for public access. However, one way to evaluate efficiency of the tool would be to compare its feedback to the one provided by human annotators. Another option would be organizing a survey for Swedish learners and asking how useful they find the feedback.

## References

Johnny Bigert, Viggo Kann, Ola Knutsson, and Jonas Sjöbergh. 2004. *Grammar Checking for Swedish Second Language Learners*, pages 33–47.

Johnny Bigert and Ola Knutsson. 2002. Robust error detection: A hybrid approach combining unsupervised error detection and linguistic knowledge. In *In Proceedings RO-MAND-02*, pages 10–19.

Luís Morgado da Costa, Roger V P Winder, Shu Yun Li, Benedict Christopher Lin Tzer Liang, Joseph Mackinnon, and Francis Bond. 2020. Automated writing support using deep linguistic parsers. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 369–377, Marseille, France. European Language Resources Association.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rickard Domeij, Ola Knutsson, Johan Carlberger, and Viggo Kann. 2000. Granska - an efficient hybrid system for swedish grammar checking. In *I Proceedings of the 12 th Nordic Conference in Computational Linguistics, Nodalida-99*.

Anisia Katinskaia, Javad Nouri, and Roman Yangarber. 2017. Revita: a system for language learning and supporting endangered languages. *Linköping electronic conference proceedings*, 134:27–35. Volume: Proceeding volume: ; Joint 6th Workshop on NLP for Computer Assisted Language Learning and 2nd Workshop on NLP for Research on Language Acquisition ; Conference date: 22-05-2017 Through 22-05-2017.

Anisia Katinskaia, Javad Nouri, and Roman Yangarber. 2018. Revita: a language-learning platform at the intersection of its and call. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 4084–4093, France. European Language Resources Association (ELRA). International Conference on Language Resources and Evaluation, LREC 2018 ; Conference date: 07-05-2018 Through 12-05-2018.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the national library of sweden – making a swedish bert.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Björn Rudzewitz, Ramon Ziai, Kordula De Kuthy, and Detmar Meurers. 2017. Developing a web-based workbook for English supporting the interaction of students and teachers. In *Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition*, pages 36–46, Gothenburg, Sweden. LiU Electronic Press.

Björn Rudzewitz, Ramon Ziai, Kordula De Kuthy, Verena Möller, Florian Nuxoll, and Detmar Meurers. 2018. Generating feedback for English foreign language exercises. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 127–136, New Orleans, Louisiana. Association for Computational Linguistics.

Jonas Sjöbergh. 2005. Faking errors to avoid making errors: Very weakly supervised learning for error detection in writing. pages 506–512.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

Elena Volodina, Ildikó Pilán, Ingegerd Enström, Lorena Llozhi, Peter Lundkvist, Gunlög Sundberg, and Monica Sandell. 2016. SweLL on the rise: Swedish learner language corpus for European reference level studies. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 206–212, Portorož, Slovenia. European Language Resources Association (ELRA).

# Term Spotting: A quick-and-dirty method for extracting typological features of language from grammatical descriptions

**Harald Hammarström**
Uppsala University
harald.hammarstrom@lingfil.uu.se

**One-Soon Her**
National Chengchi University
onesoon@gmail.com

**Marc Tang**
University Lumière Lyon 2
marc.tang@univ-lyon2.fr

## Abstract

Starting from a large collection of digitized raw-text descriptions of languages of the world, we address the problem of extracting information of interest to linguists from these. We describe a general technique to extract properties of the described languages associated with a specific term. The technique is simple to implement, simple to explain, requires no training data or annotation, and requires no manual tuning of thresholds. The results are evaluated on a large gold standard database on classifiers with accuracy results that match or supersede human inter-coder agreement on similar tasks. Although accuracy is competitive, the method may still be enhanced by a more rigorous probabilistic background theory and usage of extant NLP tools for morphological variants, collocations and vector-space semantics.

## 1 Introduction

The present paper addresses extraction of information about languages of the world from digitized full-text grammatical descriptions. For example, the below reference describes a language called Kagulu, whose grammatical properties are of interest for various linguistic predicaments.

> Petzell, Malin. (2008) *The Kagulu language of Tanzania: grammar, text and vocabulary* (East African languages and dialects 19). Köln: Rüdiger Köppe Verlag. 234pp.

The typical instances of such information-extraction tasks are so-called typological features, e.g., whether the language has tone, prepositions, SOV basic constituent order and so on, similar in spirit to those found in the database WALS `wals.info` (Dryer and Haspelmath, 2013).

Given its novelty, only a few embryonic approaches (Virk et al., 2019; Wichmann and Rama, 2019; Macklin-Cordes et al., 2017; Hammarström, 2013; Virk et al., 2017) have addressed the task so far. Of these, some are word-based and some combine words with more elaborate analyses of the source texts such as frame-semantics (Virk et al., 2019). All approaches so far described require manual tuning of thresholds and/or supervised training data.

For the present paper, we focus on the prospects of term spotting, but in a way that obviates the need for either manual tuning of thresholds or supervised training data. However, this approach is limited to the features for which a (small set of) specific terms frequently signal the presence thereof, e.g., `classifier`, `suffix(es)`, `preposition(s)`, `rounded vowel(s)` or `inverse`. Term spottting is not applicable for features which are expressed in a myriad of different ways across grammars, e.g., as whether the verb agrees with the agent in person. It may be noted that the important class of word-order features, which are among the easiest for a human to discern from a grammar, typically belong to the class of non-term-signalled features unless there is a specific formula such as SOV or N-Adj gaining sufficient popularity in grammatical descriptions. Term-signalled features are, of course, far simpler to extract, but not completely trivial, and hence the focus the present study.

The general-form premises to the problem addressed here are as follows. There is a set $D$ of raw-text descriptions of entities from a set $S$, such that each $d \in D$ mainly describes exactly one $s \in S$. If a term $k$ describing a property of objects in $S$ occurs in a document $d$ to a significant degree, the object $s$ described in $d$ actually has the property signalled by $k$. These premises apply to other domains and texts, e.g., ethnographic descriptions, than the linguistic descriptions in the present study. Judging from the surveys of Nasar et al. (2018) and Firoozeh et al. (2020), the premise that each

$d \in D$ mainly describes exactly one $s \in S$ is not dominant across scientific domains. Consequently most work has focussed on the broader tasks of extracting key-insights and salient keywords from scientific documents. We are not aware of any work in other domains on the specific task addressed in this paper.

## 2 Data

The data for the experiments in this essay consists of a collection of over 10 000 raw text grammatical descriptions digitally available for computational processing (Virk et al., 2020). The collection consists of (1) out-of-copyright texts digitized by national libraries, archives, scientific societies and other similar entities, (2) texts posted online with a license to use for research, usually by university libraries and non-profit organizations (notably the Summer Institute of Linguistics), and (3) texts under publisher copyright where quotations of short extracts are legal. For each document, we know the language it is written in (the meta-language, usually English, French, German, Spanish, Russian or Mandarin Chinese, see Table 1), the language(s) described in it (the target language, typically one of the thousands of minority languages throughout the world) and the type of description (comparative study, description of a specific feature, phonological description, grammar sketch, full grammar etc). The collection can be enumerated using the bibliographical- and metadata contained in the open-access bibliography of descriptive language data at `glottolog.org`. The grammar/grammar sketch collection spans no less than 4 527 languages, very close to the total number of languages for which a description exists at all (Hammarström et al., 2018).

| Meta-language | | # lgs | # documents |
|---|---|---|---|
| English | eng | 3497 | 7284 |
| French | fra | 826 | 1323 |
| German | deu | 620 | 813 |
| Spanish | spa | 394 | 808 |
| Russian | rus | 288 | 498 |
| Chinese | cmn | 180 | 234 |
| Portuguese | por | 141 | 274 |
| Indonesian | ind | 130 | 210 |
| Dutch | nld | 113 | 171 |
| Italian | ita | 92 | 141 |
| ... | ... | ... | ... |

Table 1: Meta-languages of the grammatical descriptions in the present collection.

Figure 1 has an example of a typical source document — in this case a German grammar of the Ewondo [ewo] language of Cameroon — and the corresponding OCR text which illustrates the typical quality. In essence, the OCR correctly recognizes most tokens of the meta-language but is hopelessly inaccurate on most tokens of the vernacular being described. This is completely expected from the typical, dictionary/training-heavy, contemporary techniques for OCR, and cannot easily be improved on the scale relevant for the present collection. However, some post-correction of OCR output very relevant for the genre of linguistics is possible and advisable (see Hammarström et al. 2017). The bottom line, however, is that extraction based on meta-language words has good prospects in spite of the noise, while extraction of accurately spelled vernacular data is not possible at present.

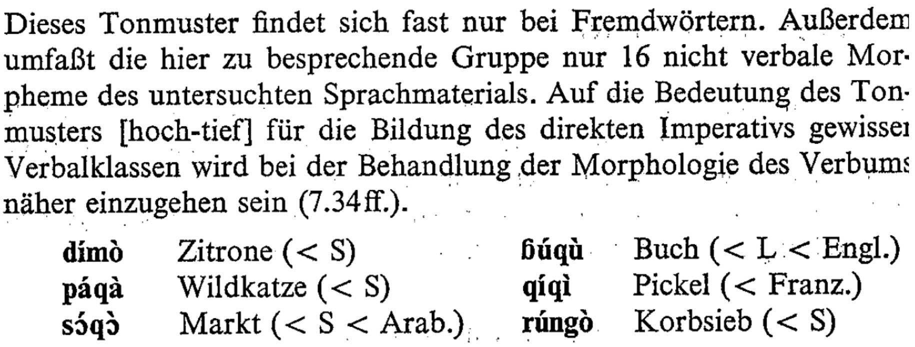
## 3 Model

At first blush, the problem might seem trivial: simply look for the existence of the term and/or its relative frequency in a document, and infer the feature associated with the term. Unfortunately, to simply look for the existence of a term is too naive. In many grammars, terms for grammatical features do occur although the language being described, in fact, does not exhibit the feature. For example, the grammar may make the explicit statement that there are "no X" incurring at least one occurrence[1]. Also, what frequently happens is that comments and comparisons are made with other languages — often related languages or other temporal stages — than the main one being described[2]. Furthermore, there is always the possibility that a term occurs in an example sentence, the title of a reference or the like. However, such "spurious" occurrences will not likely be frequent, at least not as frequent

[1] One example is the Pipil grammar of Campbell (1985, 61) which says that Pipil has no productive postpositions:

> "It should be noted that unlike Proto-Uto-Aztecan (Langacker 1977:92-3) Pipil has no productive postpositions. However, it has reflexes of former postpositions both in the relational nouns (cf. 3.5.2) and in certain of the locative suffixes (cf. 3.1.3)" (Campbell, 1985, 61).

[2] For example, Lorenzino (1998)'s description of Angolar Creole Portugues [aoa] contains a number of references to the fate of nouns that were masculine in Portuguese, yet the modern Angolar does not have masculine, or other, gender.



Dieses Tonmuster findet sich fast nur bei Fremdwörtern. Außerdem umfaßt die hier zu besprechende Gruppe nur 16 nicht verbale Morpheme des untersuchten Sprachmaterials. Auf die Bedeutung des Tonmusters [hoch-tief] für die Bildung des direkten Imperativs gewisser Verbalklassen wird bei der Behandlung der Morphologie des Verbums näher einzugehen sein (7.34ff.).

| | | | |
|---|---|---|---|
| **dímò** | Zitrone (< S) | **ɓúqù** | Buch (< L < Engl.) |
| **páqà** | Wildkatze (< S) | **qíqì** | Pickel (< Franz.) |
| **sɔ́qɔ̀** | Markt (< S < Arab.) | **rúngò** | Korbsieb (< S) |

```
Dieses Tonmuster findet sich fast nur bei Fremdwörtern. Außerdem
umfaßt die hier zu besprechende Gruppe m1r 16 nicht verbale
Morpheme des untersuchten Sprachmaterials. Auf die Bedeutun& des
Tonmus.ters [hoch-tief] für die Bildung des direkten Imperativs gewisser
Verbalklassen wird bei der Behandlung .der Morphologie des Verbums
nähereinzugehen sein (7.34ff.). Â·
Â·
Â·
dimo
paqa
s˜q;,
```

Figure 1: An example of OCR output.

as a term for a grammatical feature which actually belongs to the language and thus needs to be described properly. But how frequent is frequent enough? We will try to answer this question.

Let us assume that a full-text grammatical description consists of four classes of terms:

**Genuine descriptive terms:** Terms that describe the language in question.

**Noise terms:** Descriptive terms that do not accurately describe the language in question (i.e., through remarks on other languages or of things not present, as explained above).

**Meta-language words:** Words in the meta-language, e.g., *'the'*, *'a'*, *'run'* if the meta-language of description is English, that are not linguistic descriptive terms.

**Language-specific words:** Words that are specific to the language being described but which do not describe its grammar. These can be morphemes of the language, place names in the language area, ethnographic terms etc.

We are interested in the first class, and in particular, to distinguish them from the second class. Except for rare coincidences, the words from these two classes do not overlap with the latter two, so they can be safely ignored when counting linguistic descriptive terms. Of the terms that genuinely describe a language, we would expect their frequency distribution in a grammar to mirror their functional load (Meyerstein, 1970), i.e., their relative importance, in the language being described. Thus we assume each language has a theoretical distribution $L(t)$ of terms $t$ which is our object of interest. However, as noted, grammars typically also contain "noise" terms which distort the reflection of $L(t)$. A simple model for the frequency distribution of the terms of a grammar $G(t)$ is that it is composed merely of a sample of the "true" underlying descriptive terms $L(t)$ and a "noise" term $N(t)$, with a weight $\alpha$ balancing the two:

$$G(t) = \alpha \cdot L(t) + (1 - \alpha) \cdot N(t)$$

For example, if a language actually has duals, $L(dual) > 0$, perhaps close to 0.0 if there are only a handful of nouns with dual forms, but higher if there are dual pronouns, dual agreement, special dual case forms and so on. For most languages, we expect the functional load of verbs to be rather high. The purity level $\alpha$, captures the fraction of tokens which actually pertain to the language, as opposed to those that do not. (Those tokens are typically of great interest for the reader of the grammar — they are "noise" only from the perspective of extraction as in the present paper.)

Suppose now that we have several different grammars for the *same* language. As they are the describing the same language, their token distributions are all (independent?) samples of the *same* $L(t)$, but there is no reason to suppose the noise level and the actual noise terms to be the same across different grammars. Thus we have:

$$
\begin{aligned}
G_1(t) &= \alpha_1 \cdot L(t) + (1 - \alpha_1) \cdot N_1(t) \\
G_2(t) &= \alpha_2 \cdot L(t) + (1 - \alpha_2) \cdot N_2(t) \\
&\ldots\ldots \\
G_n(t) &= \alpha_n \cdot L(t) + (1 - \alpha_n) \cdot N_n(t)
\end{aligned}
$$

If we had infinitely many independent grammars accurately describing a language (and nothing else), their combined distribution would converge to $L(t)$ in the limit. Without the luxury of so many representative grammars, we can still attempt the simpler task of estimating the purity levels $\alpha_i$ of each grammar. That is, given actual distributions $G_1(t), \ldots, G_n(t)$ how can we make a heuristic estimate of $\alpha_i$? The following procedure suggests itself. Take each term $t$ for each grammar $G_i$ and calculate the *generality* of its incidence $g^i_L(t)$ by comparing the fraction in $G_i(t)$ to the fraction of $t$ in all other grammars for the language $L$.

$$g^i_L(t) = \frac{\frac{1}{n-1} \sum_{j \neq i} G_j(t)}{G_i(t)}$$

For example, suppose $G_i(dual) = 0.1$ for some grammar $G_i$. Maybe for two other grammars of the same language, $G_j(dual) = 0.01$ and $G_k(dual) = 0.00$, this term barely occurs. The term *dual* would then have poor generality $g^i_L(dual) = \frac{\frac{1}{2} \cdot (0.00 + 0.01)}{0.1} = 0.05$. Some real examples of the generality of a few terms are found in Cojocaru (2004)'s grammar of Romanian given five other Romanian grammars are shown in Table 2. Terms like `triphthongs`, `gender` and `stress` have a role in describing the language and consequently show a generality close to 1.0, while "noise" terms like `cojocaru` and `ghe` are less common as items of description of the Romanian language.

Grammars with lots of terms with poor generality have a high level of noise, and, conversely, grammars where all terms have a reciprocated proportion in other grammars are pure, devoid of noise. Thus, $\alpha_i$ can be gauged as:

$$\alpha_i = \frac{\sum_t g^i_L(t) \cdot G_i(t)}{\sum_t G_i(t)}$$

To remove outliers and speed up the calculation by removing hapax terms, in the experiments below, we measure all frequencies by logarithm.

We now return to the question "how frequent is frequent enough?". We can now rephrase this as: does the frequency of a term in a grammar exceed its noise level (1-$\alpha$)? Given that we know $\alpha_i$ for a grammar $G_i$, let us make the assumption that the fraction (1-$\alpha_i$) of least frequent tokens are "noise". Simply subtracting the fraction (1-$\alpha_i$) of tokens of the least frequent types effectively generates a threshold $\bar{t}$ separating the tokens being retained versus those subtracted. For example, the grammar of Romanian by Cojocaru (2004) has an $\alpha_i$ of 0.81 and contains a total of 83 365 tokens. We wish to subtract $(1 - 0.81) \cdot 83365 \approx 15839$ tokens from the least frequent types. It turns out in this grammar that this removes all the types which have a frequency of 9 or less, rendering the frequency threshold $\bar{t} = 9$.

Let us look at an example. Table 3 has a list of grammars/grammar sketches of Romanian. Each grammar has a corresponding $\alpha_i$ purity level as described above, the total number of tokens, and the frequency threshold $\bar{t}$ induced by $\alpha$ and the token distribution. The last three columns concern the terms `masculine`, `feminine` and `neuter` respectively. The cells contain the frequency of the corresponding term, as well as the fraction of pages on which it occurs. The fraction of page occurrences is, of course, similar to, and highly correlated with the fraction of tokens but is often easier to interpret intuitively. We show it here for reference, although it is not advantageous to make use of in any of the above calculations. Thus, for example, in Cojocaru (2004) the term `masculine` occurs 240 times in total, distributed onto 74 of the total 184 pages ($\approx$ 0.40). The cells with a frequency that exceeds the threshold $\bar{t}$ for their corresponding grammar are shown in green, indicating that the term in question is probably genuinely describing the language. In this case, by majority consensus, we can infer that the language Romanian [ron] does have all three of masculine, feminine and neuter.

| $t$ | cojocaru | triphthongs | gender | stress | ghe | ... |
|---|---|---|---|---|---|---|
| Cojocaru 2004 | 0.00002 | 0.00004 | 0.00052 | 0.00025 | 0.00006 | ... |
| Agard 1958 | 0.00000 | 0.00002 | 0.00012 | 0.00078 | 0.00000 | ... |
| Gönczöl-Davies 2008 | 0.00002 | 0.00015 | 0.00046 | 0.00013 | 0.00002 | ... |
| Mallinson 1986 | 0.00000 | 0.00000 | 0.00103 | 0.00036 | 0.00000 | ... |
| Mallinson 1988 | 0.00000 | 0.00000 | 0.00055 | 0.00036 | 0.00000 | ... |
| Murrell and Ştefănescu Drăgăneşti 1970 | 0.00000 | 0.00004 | 0.00042 | 0.00027 | 0.00000 | ... |
| $g_{ron}^{\text{Cojocaru 2004}}(t)$ | 0.18 | 1.20 | 0.99 | 1.51 | 0.07 | |

Table 2: Some example terms from Cojocaru (2004) and their generality $g_{ron}^{\text{Cojocaru 2004}}(t)$ given five other Romanian grammars.

Romanian [ron]

| Grammar | $\alpha_i$ | # tokens | $\bar{t}$ | masculine | feminine | neuter |
|---|---|---|---|---|---|---|
| Cojocaru 2004 | 0.81 | 83365 | 9 | 240 0.40 (74/184) | 259 0.46 (84/184) | 124 0.23 (43/184) |
| Murrell and Ştefănescu Drăgăneşti 1970 | 0.72 | 95226 | 13 | 3 0.01 (3/424) | 5 0.01 (5/424) | 4 0.01 (3/424) |
| Gönczöl-Davies 2008 | 0.68 | 45423 | 9 | 63 0.13 (30/233) | 75 0.15 (34/233) | 23 0.06 (13/233) |
| Agard 1958 | 0.68 | 51239 | 9 | 23 0.08 (10/123) | 28 0.08 (10/123) | 0 0.00 (0/123) |
| Mallinson 1988 | 0.66 | 11019 | 4 | 18 0.30 (9/30) | 18 0.23 (7/30) | 18 0.17 (5/30) |
| Mallinson 1986 | 0.82 | 105018 | 6 | 119 0.15 (57/375) | 110 0.12 (46/375) | 25 0.03 (11/375) |
| Majority consensus | | | | True | True | True |

Table 3: Example grammars of Romanian and the frequencies of the terms masculine, feminine and neuter.

## 4 Evaluation

Thanks to a large manually elaborated database of languages with classifiers[3] (Her et al., 2021) we were able to do a formal evaluation of extraction accuracy for this feature. We extracted the feature `classifier(s)` from 7 284 grammars/grammar sketches written in English spanning 3 220 languages. Each language was assessed as per the majority vote of the extraction result of each individual description, with ties broken in favour of a positive result. For languages where only one description exists, the noise-level was taken to be the average noise-level of grammars of other languages of similar size (as measured by number of tokens).

| Gold Standard | Term-Spotting | # languages |
|---|---|---|
| False | False | 2 357 |
| True | True | 512 |
| True | False | 317 |
| False | True | 34 |
| | | 3 220 |

Table 4: Evaluation of term-spotting against a Gold Standard database of classifier languages.

A comparison between the Gold Standard database and the extracted data is shown in Table 4. The overall accuracy is 89.1%, to be compared with human inter-coder agreement on similar tasks, i.e., 85.9% or lower (as per Donohue 2006 and Plank 2009, 67-68). Not surprisingly, the method has better precision ($\frac{512}{512+34} \approx 0.94$) than it has recall ($\frac{512}{512+317} \approx 0.62$). The majority of errors are languages with classifiers which are not recognized as such by the term-spotting technique. Simple inspection reveals that in the majority of these cases, a different term, e.g., "enumerative" is used in place of the term in question. There are also errors where the automatic technique infers a slightly too high threshold for languages which have grammars from a large temporal range. The fact that descriptive tradition changes over time may be reason to refine the procedure for calculating reciprocated proportions.

We may add a few remarks on some obvious refinements. Excluding negative polarity mentions, by which we mean mentions where `no|not|absent|absence|absense|lack|neither|nor|cannot` occurs in the same sentence as the sought-after term, make no significant change to the overall accuracy. Using the temporally latest description only (instead of a majority vote) to assess the status for a language with several grammars, also made no significant change to the overall accuracy (in fact, it decreased by 2 percentage points). Furthermore, using the most extensive description only, i.e., the longest grammar or longest grammar sketch if there are no full grammars, had a negative impact on overall accuracy (down by 8 percentage points). These results seem to speak in favour of making use of multiple witnesses for each language if they are available, even if they are of different lengths and ages. If these impressions generalize, length and age differences between grammars — which are real — need to be addressed in a more sophisticated manner than simply excluding the old and short.

The above evaluation is relevant for the case when there is a specific term (or an enumerable set thereof) associated with the desired feature. It then shows what accuracy one may expect without supplying a threshold or any other information than the keyword itself. Choosing the right term(s) for a given linguistic feature requires knowledge of the feature and the way is it often (not) manifested in the literature (cf. Kilarski 2013 on classifiers versus other kinds of nominal classification).

## 5 Conclusion

We have described a novel approach to the extraction of linguistic information from descriptive grammars. The method requires only a term of interest, but no manual tuning of thresholds or annotated training data. However, the approach can only address information that is associated with an enumerable set of specific terms. When this is the case, a broad evaluation shows that the results match or exceed the far more time-consuming manual curation by humans. Future work includes automated handling of collocations and morphological variants, vector-space lexical semantics, automated multi-lingual extraction and establishing the method on more rigorous probabilistic theory.

## Acknowledgments

This research was made possible thanks to the financial support of the From Dust to Dawn: Multilingual Grammar Extraction from Grammars project funded by Stiftelsen Marcus och Amalia Wallen-

[3] See Aikhenvald (2000) and references therein for issues surrounding the definition of this feature.

bergs Minnesfond 2017.0105 awarded to Harald Hammarström (Uppsala University) and the Dictionary/Grammar Reading Machine: Computational Tools for Accessing the World's Linguistic Heritage (DReaM) Project awarded 2018-2020 by the Joint Programming Initiative in Cultural Heritage and Global Change, Digital Heritage and Riksantikvarieämbetet, Sweden. The second author O.-H. Her gratefully acknowledges the financial support of Taiwan's MOST Research Grants 106-2410-H-029-077-MY3 and 108-2410-H-029-062-MY3. The last author is thankful for the support of grants from the Université de Lyon (ANR-10-LABX-0081, NSCO ED 476), the IDEXLYON Fellowship (2018-2021, 16-IDEX-0005), and the French National Research Agency (ANR-11-IDEX-0007).

## References

Agard, Frederick B. 1958. A structural sketch of Rumanian. *Language*, 34(3):7–127. Language Dissertation No. 26.

Aikhenvald, Alexandra Y. 2000. *Classifiers: A Typology of Noun Categorization Devices*. Oxford Studies in Typology and Linguistic Theory. Oxford: Oxford University Press, Oxford.

Campbell, Lyle. 1985. *The Pipil Language of El Salvador*, volume 1 of *Mouton Grammar Library*. Berlin: Mouton de Gruyter.

Cojocaru, Dana. 2004. *Romanian Grammar*. Durham: SEELRC.

Donohue, Mark. 2006. Review of the the world atlas of language structures. *LINGUIST LIST*, 17(1055):1–20.

Dryer, Matthew S. and Martin Haspelmath. 2013. The world atlas of language structures online. Leipzig: Max Planck Institute for Evolutionary Anthropology (Available online at http://wals.info, Accessed on 2015-10-01.).

Firoozeh, Nazanin, Adeline Nazarenko, Fabrice Alizon, and Béatrice Daille. 2020. Keyword extraction: Issues and methods. *Natural Language Engineering*, 26:259–291.

Gönczöl-Davies, Ramona. 2008. *Romanian: an essential grammar*. New York: Routledge, New York.

Hammarström, Harald. 2013. Three approaches to prefix and suffix statistics in the languages of the world. Paper presented at the Workshop on Corpus-based Quantitative Typology (CoQuaT 2013).

Hammarström, Harald, Thom Castermans, Robert Forkel, Kevin Verbeek, Michel A. Westenberg, and Bettina Speckmann. 2018. Simultaneous visualization of language endangerment and language description. *Language Documentation & Conservation*, 12:359–392.

Hammarström, Harald, Shafqat Mumtaz Virk, and Markus Forsberg. 2017. Poor man's OCR post-correction: Unsupervised recognition of variant spelling applied to a multilingual document collection. In *Proceedings of the Digital Access to Textual Cultural Heritage (DATeCH) conference*, pages 71–75. Göttingen: ACM.

Her, One-Soon, Harald Hammarström, and Marc Allassonnière-Tang. 2021. Introducing WACL: The world atlas of classifier languages. *Submitted*, page 15pp.

Kilarski, Marcin. 2013. *Nominal classification: A history of its study from the classical period to the present*. Amsterdam: John Benjamins.

Lorenzino, Gerardo A. 1998. *The Angolar Creole Portuguese of São Tomé: Its Grammar and Sociolinguistic History*. Ph.D. thesis, City University of New York.

Macklin-Cordes, Jayden L., Nathaniel L. Blackbourne, Thomas J. Bott, Jacqueline Cook, T. Mark Ellison, Jordan Hollis, Edith E. Kirlew, Genevieve C. Richards, Sanle Zhao, and Erich R. Round. 2017. Robots who read grammars. Poster presented at CoEDL Fest 2017, Alexandra Park Conference Centre, Alexandra Headlands, QLD.

Mallinson, Graham. 1986. *Rumanian*. Croom Helm Descriptive Grammars. London: Croom Helm.

Mallinson, Graham. 1988. Rumanian. In Martin Harris and Nigel Vincent, editors, *The Romance Languages*, pages 391–419. London: Croom Helm.

Meyerstein, R. S. 1970. *Functional Load: Descriptive Limitations Alternatives of Assessment and Extensions of Application*. The Hague: Mouton.

Murrell, Martin and Virgiliu Ştefănescu Drăgăneşti. 1970. *Romanian*. Teach Yourself Books. London: English Universities Press.

Nasar, Zara, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2018. Information extraction from scientific articles: a survey. *Scientometrics*, 117:1931–1990.

Plank, Frank. 2009. WALS values evaluated. *Linguistic Typology*, 13(1):41–75.

Virk, Shafqat Mumtaz, Lars Borin, Anju Saxena, and Harald Hammarström. 2017. Automatic extraction of typological linguistic features from descriptive grammars. In Kamil Ekštein and Václav Matoušek, editors, *Text, Speech, and Dialogue: 20th International Conference, TSD 2017, Prague, Czech Republic, August 27-31, 2017, Proceedings*, volume 10415 of *Lecture Notes in Computer Science*, pages 111–119. Berlin: Springer.

Virk, Shafqat Mumtaz, Harald Hammarström, Markus Forsberg, and Søren Wichmann. 2020. The DReaM corpus: A multilingual annotated corpus of grammars for the world's languages. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 871–877. Marseille, France: European Language Resources Association, Marseille, France.

Virk, Shafqat Mumtaz, Azam Sheikh Muhammad, Lars Borin, Muhammad Irfan Aslam, Saania Iqbal, and Nazia Khurram. 2019. Exploiting frame-semantics and frame-semantic parsing for automatic extraction of typological information from descriptive grammars of natural languages. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, page 1247–1256. Varna, Bulagaria: NCOMA Ltd.

Wichmann, Søren and Taraka Rama. 2019. Towards unsupervised extraction of linguistic typological features from language descriptions. First Workshop on Typology for Polyglot NLP, Florence, Aug. 1, 2019 (Co-located with ACL, July 28-Aug. 2, 2019).

# Implant Term Extraction from Swedish Medical Records – Phase 1: Lessons Learned

**Oskar Jerdhaf**[1], **Marina Santini**[2], **Peter Lundberg**[3,4], **Anette Karlsson**[3,4], **Arne Jönsson**[1]
[1] Department of Computer and Information Science, Linköping University, Sweden
oskje724@student.liu.se|arne.jonsson@liu.se
[2] RISE, Digital Health, Sweden
marina.santini@ri.se
[3] Center for Medical Image Science and Visualization (CMIV), Linköping University, Linköping, Sweden
[4] Department of Medical Radiation Physics and Department of Health, Medicine and Caring Sciences
Linköping University, Linköping, Sweden
Peter.Lundberg@liu.se|Anette.k.karlsson@regionostergotland.se

## Abstract

We present the case of automatic identification of "implant terms". Implant terms are specialized terms that are important for domain experts (e.g. radiologists), but they are difficult to retrieve automatically because their presence is sparse. The need of an automatic identification of implant terms spurs from safety reasons because patients who have an implant may be at risk if they undergo Magnetic Resonance Imaging (MRI). At present, the workflow to verify whether a patient could be at risk of MRI side-effects is manual and laborious. We claim that this workflow can be sped up, streamlined and become safer by automatically sieving through patients' medical records to ascertain if they have or have had an implant. To this aim we use BERT, a state-of-the-art deep learning algorithm based on pre-trained word embeddings and we create a model that outputs *term clusters*. We then assess the linguistic quality or term relatedness of individual term clusters using a simple intra-cluster metric that we call *cleanliness*. Results are promising.

## 1 Introduction

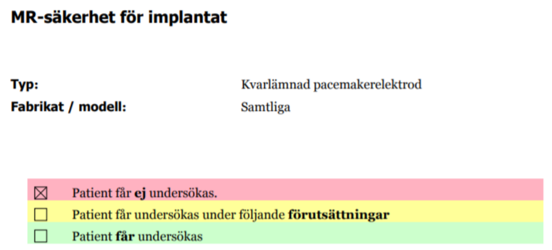
Domain-specific terminology extraction is an important task in a number of areas, such as knowledge base construction (Lustberg et al., 2018), ontology induction (Sazonau et al., 2015) or taxonomy creation (Šmite et al., 2014).

We present experiments on an underexplored type of terminology extraction that we call "focused terminology extraction". With this expression we refer to terms or to a nomenclature that represent a specialized semantic field. The automatic identification and extraction of this kind of nomenclature are a common need in many domains, e.g. medicine, dentistry, chemistry, aeronautics, engineering and the like.

In these experiments, we explore focused terminology related to the semantic field of terms that indicate or suggest the presence of "implants" in electronic medical records (EMRs) written in Swedish. More specifically, the aim of our experiments is to investigate whether it is possible to discover implant terms or implant-related words unsupervisely, i.e. learning from unlabelled data. This task is currently part of an ongoing project carried out together with LIU University Hospital. We present here the results and the lessons learned from Phase 1 of the project.

Implant terms are domain-specific words indicating artificial artefacts that replace or complement parts of the human body. Common implants are devices such as 'pacemaker', 'shunt', 'codman', 'prosthesis' or 'stent'.

The need of an automatic identification of implant terms spurs from safety reasons because patients who have an implant may or may be not submitted to Magnetic Resonance Imaging (MRI). MRI scans are very safe and most people are able to benefit from it. However, in some cases an MRI scan may not be recommended. Before undergoing an MRI scan, the following conditions must be verified: (a) the presence of metal in the body and (b) being pregnant or breastfeeding. Implants are often metallic objects, therefore it is important to know if a patient has an implant, because MRI-scanning is incompatible with some implants (e.g. the 'pulmonary artery catheter') or maybe partially compatible with some of them (e.g. the 'mitraclip'). An example of a recommendation on implants is shown in Figure 1. The translated (narrative) version of the recommendation reads: "If a pacemaker electrode is present in the patient's body, then the patient cannot be exposed to MRI scanning".



**MR-säkerhet för implantat**

| | |
|---|---|
| **Typ:** | Kvarlämnad pacemakerelektrod |
| **Fabrikat / modell:** | Samtliga |

☒ Patient får **ej** undersökas.

☐ Patient får undersökas under följande **förutsättningar**

☐ Patient **får** undersökas

Figure 1: According to this recommendation, a patient having a pacemaker electrode in the body cannot undergo MRI scanning.

Unsafe implants must be considered before MRI-scanning, as they may be contraindicative, while conditional implants can be left in the patient's body, if conditions are appropriately accounted for. One of the safety measures in MRI-clinics is to ask patients whether they have or have had an implant. This routine is not completely reliable, because a patient (especially if elderly) might have forgotten about the presence of implants in the body. When a patient has or is suspected to have an implant, the procedure of recognition and acknowledgement is manual, laborious and involves quite many human experts with specialized knowledge. The workflow of the current procedure is shown in Figure 2 and described in (Kihlberg and Lundberg, 2019).

Even if implants have been removed, metallic or electronic parts (like small electrodes or metallic clips) may have been overlooked and left *in situ*, without causing harm to patient's health before the MRI. Normally, referring physicians may be aware of the limitation of specific implants, and prior to an MRI examination, they should go through the patient's medical history by reading EMRs.

EMRs are digital documents, but the information they contain is not structured or organized in a way that makes it trivial to find implant terms quickly and efficiently. This downside can be addressed by automatically trying to identify the terms from the EMR based on their contextual usage, e.g. using word embeddings. In our experiments, we use BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019), which is the state-of-the art in computational linguistics and deep learning for several downstream tasks, e.g. text classification, question-answering or natural language understanding. Our downstream task is to find as many validated instances of implant-related words as possible in free-text EMRs. Here we present the lessons we have learned from Phase 1 of the project.

## 2 Related Work

"Focused" terminology extraction refers to mentions of a relatively small number of technical terms. From a semantic perspective, focused terminology extraction is challenging because the task implies an unsupervised discovery of a handful of specialized terms scattered in millions of words across unstructured textual documents, such as EMRs. This characterization has some similarities with the "relevant but sparse" definition in Ittoo and Bouma (2013). EMRs are written by physicians who typically use a wide range of medical sublanguages that are not only based on regular medical jargon, but also include unpredictable word-shortening and abbreviations, spelling variants of the same word (including typos), numbers, and the like. What is more, these sublanguages vary across hospitals and clinics.

Focused terminology extraction is still underexplored. Little work exists on this task, although its usefulness in real-world applications is extensive.

Recent studies exist however on general medical synonym discovery. For instance, Schumacher and Dredze (2019) compare eight neural models on the task of finding disorder synonyms in English clinical free text. In their evaluation, ELMO models performs moderately better than the other models. Before the neural revolution and the word embeddings paradigm, models for synonym extraction have been proposed for many languages and also specifically for the Swedish language. The models for Swedish presented in Henriksson et al. (2012) are based on (by now) traditional word space models, namely Random Indexing and Random Permutation. The models were designed to identify both synonyms and abbreviations. These models were built on the Stockholm EPR Corpus (Dalianis et al., 2009) and synonym extraction was evaluated on the Swedish version of MeSH and its extension[1]. Results were encouraging, but limited to terms included in Swedish MeSH, which does not cover the whole medical terminology and, what is more, does not include graphical variations that are present in the informal medical sublanguage often used in medical records.

Focussed terminology extraction could be interpreted as a special case of Named Entity Recog-

[1] https://mesh.kib.ki.se/

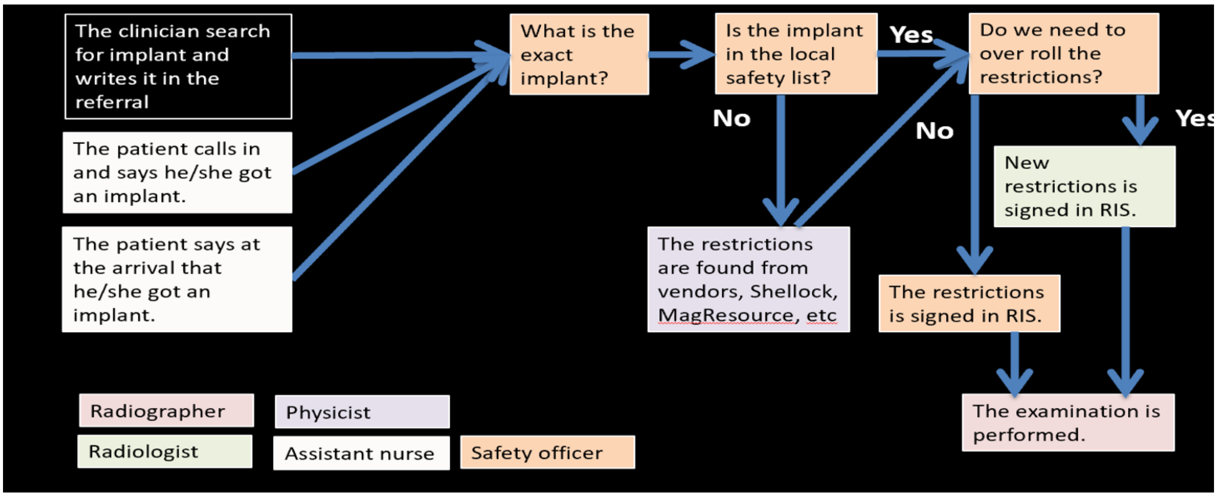

The clinician search for implant and writes it in the referral
The patient calls in and says he/she got an implant.
The patient says at the arrival that he/she got an implant.
What is the exact implant?
Is the implant in the local safety list?
Yes
No
Do we need to over roll the restrictions?
No
Yes
New restrictions is signed in RIS.
The restrictions are found from vendors, Shellock, MagResource, etc
The restrictions is signed in RIS.
The examination is performed.
Radiographer
Physicist
Radiologist
Assistant nurse
Safety officer

Figure 2: Current workflow (Kihlberg and Lundberg, 2019)

nition (NER), where the entities to be identified are words indicative of implants. We considered the possibility of fine-tuning a BERT pre-trained model on a labelled corpus of implant terms using custom labels. However, at present, this annotation endevour cannot be undertaken because it requires financial resources and a time span that are not available at the time of this publication.

We then explored the unsupervised NER solution based on BERT proposed in an article by Ajit Rajasekharan[2]. This article describes a fully-unsupervised approach to NER based on the pre-trained *bert-large-cased* (English). The approach relies on signatures indicating entities, on a morphological tagger and on BERT's Masked Language Model (MLM) head to predict candidate words for the masked positions. To put it simply, the approach combines NER and MLM using the head of the MLM to extract entities. Results seem to be promising for the NER task. However, the adaptation of this approach to the recognition of implant terms and to our domain-specific data resulted overly complex. One main hinder to this adaptation is the use of a morphological tagger. Our data is noisy and specialized and the result of a tagger on this data is certainly unreliable without a proper retraining of the tagger itself for domain and genre. Another difficult step to adapt to our task is the creation of signatures that are then handled at raw word embedding level. As the author puts it, the unsupervised NER approach works because: "BERT's raw word embeddings capture useful and separable information (distinct histogram tails with less than 0.1 % of vocabulary) about a term using other words in BERT's vocabulary. [...] The transformed versions of these embeddings output by a BERT model with MLM head are used to make predictions of masked words. The predictions also have a distinct tail. This is used to choose the context sensitive signature for a term.". First of all, the extraction of the signatures would be redundant in our case since we have already a list of implant terms (i.e the glossaries described in Section 4.1.3) automatically extracted from existing official documentation and used as search terms. Second, many of these terms are not, presumably, in the general-purpose vocabulary of the pre-trained BERT, since they are very specialized. Essentially, we ended up with the conclusion that it would be very time-consuming (if ever possible) to implement some of the steps in the fully-unsupervised BERT-NER model. The approach remains indeed inspiring and could work better or streamlined if we could re-train BERT on our domain-specific corpus, an operation that we are unable to carry out at the time of this publication.

We also explored the possibility of using a BERT model specifically fine-tuned on our corpus to predict *masked tokens* to find candidate implant terms. However, we realized that such an approach is dwarfed, because only the words in the vocabulary of the pre-trained BERT model would be suggested. If a term, e.g. "shunt", is not in that vocabulary or cannot be reconstructed using BERT tokens, it will never be "discovered" as implant term and will remain undetected.

In the experiments presented here we build on re-

[2] https://towardsdatascience.com/unsupervised-ner-using-bert-2d7af5f90b8a (published 2020, updated 2021, retrieved 2021.)

search carried out at Linköping University in close cooperation with Linköping University Hospital. Kindberg (2019) started this exploration and relied on Word2Vec (Mikolov et al., 2013). In his experiments, carried out on EMRs belonging to the cardiology clinic (see section 3), Kindberg (2019) evaluated 500 terms, i.e. 10 search words and their 50 closest neighbours. For the evaluation, all the terms were divided into 14 categories, and only three of these categories contained words indicative of implants. All in all, 26.2% of the 500 analysed words were considered words indicative of implants, i.e. "synonyms, semantically similar terms, abbreviations, misspelled terms" (p. 13).

For the same task on the same cardiology clinic, Nilsson et al. (2020) used Swedish BERT (see Section 4.1). The results presented in Nilsson et al. (2020) showed that "[o]ut of the 148 evaluated queries, 68 query words (46%) in their given context were considered to be clearly indicative for implants or other harmful objects. 27 query words (18%) were considered possibly indicative and 53 query words (36%) were considered non-indicative. For each query that was clearly or possibly indicative, five contextually similar words were identified which resulted in 475 additional words in given contexts. Among these 475 additional words, 83 (17,5%) words were considered as clearly indicative in their context, 105 (22%) were considered as possibly indicative and 287 (60,5%) were considered non-indicative. 40% of the 475 additional words identified with the KD-Tree queries and BERT were deemed to be possibly indicative or clearly indicative of implants or other harmful objects." (p.23-24).

It must be noticed that the results by Kindberg (2019) and by Nilsson et al. (2020) are not directly comparable between them since the evaluation methods are different. Although we learned a lot from these two previous studies, we are unable to compare our results with theirs, because in our experiments we create a model on two clinics, i.e. cardiology and neurology, rather than only on cardiology. What is more, our evaluation methods and metrics differ from those utilized by Kindberg (2019) and Nilsson et al. (2020).

## 3 Data: Electronic Medical Records

The data used in our experiments is the text of EMRs from two very different clinics at Linköping University Hospital, namely the cardiology clinic and the neurology clinic. The EMRs span over the latest five years and amount to about 1 million EMRs, when taken individually, and about 48000 when groped by unique patient (the breakdown of record distribution in shown in Table 1). These EMRs vary greatly in length, from just a few words to hundreds of words. This data has not yet been fully anonymised, therefore we are unable to release the datasets at the time of this publication. However, we will distribute secondary linguistic data, such as automatically created wordlists on the project website.

| Clinics | Words | SingleEMRs | GroupedEMRs |
|---|---|---|---|
| Cardiology | 45 780 055 | 664 821 | 34 044 |
| Neurology | 25 440 484 | 314 669 | 14 526 |
| Total | 71 220 539 | 979 490 | 48 088 |

Table 1: Number of words and EMRs per clinic.

## 4 Method: BERT

Previous methods to represent features as vectors were unable to capture the context of individual words in the texts, sometimes leading to a poor representation of natural language. When using a traditional text classifier, one of the simplest ways to represent text is to use bag-of-words (BOW), where each word (feature) in the text is stored together with their relative frequency, ignoring word position of the word in the sentence and in the text. A more advanced way to represent features is by using word embeddings, where each feature is mapped to a vector of numbers. The pioneer of this approach was a method called Word2Vec (Mikolov et al., 2013). A big leap forward was achieved with BERT (Bidirectional Encoder Representations from Transformers), which uses a multi-headed self-attention mechanism to create deep bidirectional feature representations, able to model the whole context of all words in a sequence. Bidirectional refers to the ability of simultaneously learning left and right word context. Up to BERT, bidirectionality could be achieved only by modeling two separate networks for each direction that would later be combined, as in (Peters et al., 2018). A BERT model uses a transfer learning approach, where it is pre-trained on a large amount of data. After learning deep bidirectional representations from unlabelled text, BERT can be further finetuned for several downstream tasks.

BERT is a powerful but complex model. According to the Occam's razor principle, simplicity must be preferred whenever possible. To comply to this principle, we carried out a few preliminary experiments on samples taken from the current dataset (cardiology + neurology) with approaches less complex than BERT, like distributional semantics based on BOW[3] and Word2Vec[4]. Results on the samples showed that BERT performed better than the others methods. A comparative study on the whole dataset (not only samples) is in preparation. These results, together with additional experiments that are still in progress, will also be available in the final project's report that will be handed in to the funding body.

In the experiments presented here, we fine-tuned BERT for focussed terminology extraction and relied on PyTorch (an open source machine learning framework[5]) (Paszke et al., 2019) and used the Huggingface transformers library for BERT (Wolf et al., 2019) available and ready to use[6].

## 4.1 Swedish BERT

### 4.1.1 Pre-Trained Model

The pre-trained BERT model used in these experiments is the *bert-base-swedish-cased* released by The National Library of Sweden (Malmsten et al., 2020)[7]. To provide a representative BERT model for the Swedish language, the model was trained on approximately 15-20 gigabyte of text (200M sentences, 3000M tokens) from a range of genres and text types including books, news, and internet forums. The model was trained with the same hyperparameters as first published by Google and corresponded to the size of Google's base version of BERT with 12 so-called transformer blocks (number of encoder layers), 768 hidden units, 12 attention heads and 110 million parameters.

A BERT model has a predefined vocabulary. This vocabulary is a set of words known to the model and it is used to tokenize words. A token can in this case be a common word, a common subpart of a word or a single letter. Each object in the vocabulary of the model has a known embedding. To use the model for finding the embedding of a new word the model was used to tokenize the word, which means that it would try to rebuild the word using as few tokens from the vocabulary as possible. The pre-trained BERT model used in this study had a vocabulary of 50325 words. Pre-trained model hyperparameters are listed in Table 2.

| Hyperparemeter | Dimensions/Value |
|---|---|
| Dropout | 0.1 |
| Hidden Activation | GELU |
| Hidden Size | 768 |
| Embedding Size | 512 |
| Attentional Heads | 12 |
| Hidden Layers | 12 |
| Forward Size | 3072 |
| Vocabulary Size | 50325 |
| Trainable Parameters | $11 \cdot 10^7$ |

Table 2: Pre-training parameters

### 4.1.2 Fine-Tuning the Pre-Trained Model: Phase 1

We call this tine-tuning "Phase 1" because in the near future we are going to try out different fine-tuning configurations in order to understand how to determine the optimal hyperparameters' settings for the task at hand. In Phase 1, the decisions about how to set parameters were made partly based on the original BERT paper (Devlin et al., 2019), partly on previous findings based on electronic health records notes (Li et al., 2019), partly on the observation of our current data. Hyper-parameters used for fine-tuning in this study are shown in Table 3. We relied on the Adam algorithm with default values for its hyperparameters as indicated by (Kingma and Ba, 2014). The pre-processed EMRs and the pre-trained model were fed into a Python script.

| Hyperparameter | Dimension/Value |
|---|---|
| Epochs | 3 |
| Batch Size | 32 |
| Block Size | 64 |
| Learning Rate | $5e-5$ |

Table 3: Parameters used for fine-tuning

The model was fine-tuned with MLM (Masked Language Model), a technique which allows bidirectional training. MLM consists in replacing 15% of the words in each sequence with a [MASK] token before feeding word sequences into BERT. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence. The block size was set to 64, which means that

[3] To find synonyms or semantically related words, the *textstat_simil* function of the Quanteda R package (Benoit et al., 2018) was used.

[4] Package 'word2vec, R wrapper, https://cran.r-project.org/web/packages/word2vec/word2vec.pdf

[5] https://pytorch.org/

[6] https://huggingface.co/transformers/

[7] https://github.com/Kungbib/swedish-bert-models

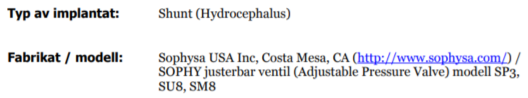
sequences with fewer than 64 tokens are padded to meet this length, and sequences with more than 64 tokens are truncated. Actually, the value of 64 is generous since according to our current calculations the average sentence length in tokens is 12. The fine-tuning took approximately 15 hours per clinic to complete using the computing resources shown in Table 4.

| Label | Description |
|---|---|
| CPU | Intel Xeon - 12x(E5-2620 v3) |
| GPU | NVIDIA Quadro M4000<br>[8GB(VRAM)\|20GB(Shared)] |
| Clock Speed | 2.40GHz |
| Memory (RAM) | 40GB |

Table 4: Details of computing resources.

### 4.1.3 Discovering Contextually-Similar Implant Terms

We used the MRI-safety handbook (SMRlink) available at the hospital website to automatically create glossaries of implants or implant-related terms. In these experiments, we used two glossary versions, an extended version containing 753 terms that include some noise, i.e. non-implant terms, and a baseline version containing 461 terms and less noise, but also fewer terms. The extended glossary was automatically built from several sections of the documents that can be found in SMRlink. The baseline version was extracted only from the headings ”Typ av implatat” and ”Fabrikat / modell” (see Figure 3). The advantage of the extended version is the presence of potentially more implant terms. Neither of the two glossary versions was validated by domain experts, since we wanted to limit human intervention as much as possible and explore the effect of different choices.



| | |
|---|---|
| **Typ av implantat:** | Shunt (Hydrocephalus) |
| **Fabrikat / modell:** | Sophysa USA Inc, Costa Mesa, CA (http://www.sophysa.com/) / SOPHY justerbar ventil (Adjustable Pressure Valve) modell SP3, SU8, SM8 |

Figure 3: The terms in the baseline glossary were extracted from the headings shown in this picture.

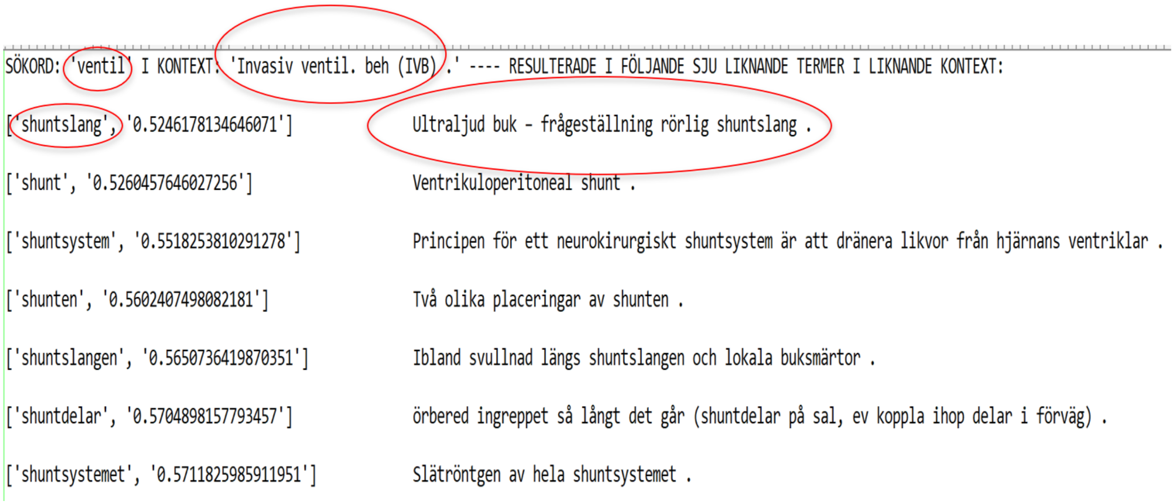
With glossary terms and the corpus, queries were created. A query is basically an example sentence containing a glossary term. Our queries are randomly chosen in the corpus. The model retrieves sentences similar to the queries and extract the term that is most similar to the glossary term that the queries exemplify (see Figure 4).

In this paper, we present the results of a BERT model evaluated using 15 queries for each glossary term. The queries were randomly chosen and used to find contextually similar sentences. This BERT model first identifies sentences in the corpus that are similar to the queries, then it extracts words in the BERT “discovered” sentences that have similar syntactic/semantic role/slot (i.e. the same “semantic role” in a broad sense) as the glossary terms that were used to build the queries. Since our corpus is sizeable, we decided that pairwise cosine similarity metric (brute force) would have been too inefficient with ordinary computing resources, and not compliant to the Green NLP paradigm (Derczynski, 2020). To build the search space we used instead the scikit-learn implementation of the KD-Tree and BallTree algorithms (Pedregosa et al., 2011), both with default distance metrics. KDTree (short for k-dimensional tree) is a binary space partitioning data structure for organizing points in a k-dimensional space and it is useful when using multidimensional search key (e.g. range searches and nearest neighbour searches[8]). While the KDTree is “a binary tree structure which recursively partitions the parameter space along the data axes, dividing it into nested orthotropic regions into which data points are filed”, BallTrees “partition[s] data in a series of nesting hyper-spheres. This makes tree construction more costly than that of the KD tree, but results in a data structure which can be very efficient on highly structured data, even in very high dimensions”. KDTree and BallTree are both memory-intensive. In order to speed up this part of the computation, the data was split into chunks. Each individual chunk was used to generate results for all queries and then the most contextually similar words and sentences across all of the chunks were selected for the final results. The results used in this paper were generated with chunks of 50000 tokenized sentences at the time.

## 4.2 Evaluation

To judge whether a term discovered using this BERT model is indicative of the presence of implants, special domain knowledge is required. In some cases, it may be obvious that a term indicates implants. In other cases, it may be less obvious due to very domain-specific sublanguage. For this reason, manual evaluation of BERT discoveries was carried out by two MRI-physicists from the

[8] https://scikit-learn.org/stable/modules/neighbors.htm



```
SÖKORD: 'ventil' I KONTEXT: 'Invasiv ventil. beh (IVB).' ---- RESULTERADE I FÖLJANDE SJU LIKNANDE TERMER I LIKNANDE KONTEXT:

['shuntslang', '0.5246178134646071']          Ultraljud buk - frågeställning rörlig shuntslang .

['shunt', '0.5260457646027256']               Ventrikuloperitoneal shunt .

['shuntsystem', '0.5518253810291278']         Principen för ett neurokirurgiskt shuntsystem är att dränera likvor från hjärnans ventriklar .

['shunten', '0.5602407498082181']             Två olika placeringar av shunten .

['shuntslangen', '0.5650736419870351']        Ibland svullnad längs shuntslangen och lokala buksmärtor .

['shuntdelar', '0.5704898157793457']          örbered ingreppet så långt det går (shuntdelar på sal, ev koppla ihop delar i förväg) .

['shuntsystemet', '0.5711825985911951']       Slätröntgen av hela shuntsystemet .
```

Figure 4: A mock-up of the results retrieved by a query: 'Sökord' (en: search term) is a glossary term. 'I kontext' (en: in context) is a query where the search term appears. The model extract 7 sentences similar to the query and extracts terms contexually similar to the glossary term.

Radiology clinic at Linköping University Hospital, who assessed independently the terms discovered by the BERT model. For the evaluation with used the results obtained with KDTree and the extended glossary, which amount to 4636 BERT terms. More specifically, we started up with 753 glossary terms (unigrams) including noise; for each glossary term, a set of 15 queries was created (15 is an arbitrary choice); KDTree was used to search the vector space from which we extracted 7 nearest neighbours for a given query (7 is an arbitrary choice) (see Figure 4); then we merged the results for all the queries together and removed duplicates.

The two MRI-physicists received an excel file containing the list of terms to be assessed without any context, and short instructions. They were instructed to judge whether the term can give an indication that the patient has or has had an implant. They were asked to use the following ratings on a three-degree scale: **Y** = *yes, it gives me an indication that the patient has or has had an implant*; **N** = *No, it DOES NOT give me any indication that the patient has or has had an implant*; **U** = unsure, *the term could or could not give me an indication of an implant, but I cannot decide without more context*. The inter-rater agreement was then computed on their judgements. Results are presented in the next section.

## 5 Results and Evaluation

**Inter-Rater Agreement**. We measured the inter-rater agreement between the two MRI-physicists by using percentage (i.e. the proportion of agreed upon documents in relation to the whole without chance correction), the classic unweighted Cohen's kappa (Cohen, 1960) and Krippendorff's alpha (Krippendorff, 1980) to get a straightforward indication of the raters' tendencies.

Cohen's kappa assumes independence of the two coders and is based on the assumption that "if coders were operating by chance alone, we would get a separate distribution for each coder" (Artstein and Poesio, 2008). This assumption intuitively fits our expectations. Krippendorff's alpha is similar to Cohen's kappa, but it also takes into account the extent and the degree of disagreement between raters (Artstein and Poesio, 2008).

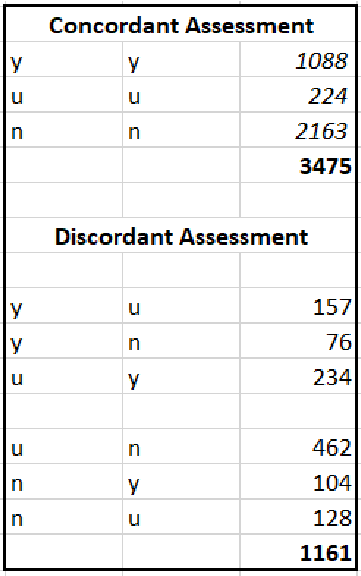
| Terms | Percentage | Cohen's Kappa | Krippendorff's Alpha |
|---|---|---|---|
| 4636 | 75% | 0.575 | 0.573 |

Table 5: Inter-rater agreement on 4636 BERT terms.

| Rater | Y | N | U |
|---|---|---|---|
| Rater-1 | 1 426 (30.8%) | 2 701 (58.2%) | 509 (11%) |
| Rater-2 | 1 321 (28.5%) | 2 395 (51.5%) | 920 (20%) |

Table 6: Breakdown by rater.

Tables 5 and 6 show the breakdown of the inter-rater agreement of the 4636 terms discovered by BERT. The raters agree on 3475 terms, of which **1088** were assessed to be indicative implant terms (approx. 23.5%), 2163 terms were assessed not to be indicative of implants, and for 224 terms both raters agreed on being "unsure". The raters disagreed on 1161 terms. This means that BERT helped discover 75% of terms on which the two raters are concordant (i.e. 1088+2163+224), and 25% on which they are discordant (see Figure 5). Out of the 1088 BERT terms indicative of implants, about 900 were not in the extended glossary and more than 1000 were not present in the baseline glossary, e.g. 'carillon-device' or 'cochlea'. Therefore these BERT terms make a useful addition to the glossaries. Out of 2163 non-indicative BERT terms, about 2000 were not in the glossaries, which suggests that the level of noise in the glossaries is relatively small.



| | | |
|---|---|---|
| **Concordant Assessment** | | |
| y | y | *1088* |
| u | u | *224* |
| n | n | *2163* |
| | | **3475** |
| | | |
| **Discordant Assessment** | | |
| | | |
| y | u | 157 |
| y | n | 76 |
| u | y | 234 |
| | | |
| u | n | 462 |
| n | y | 104 |
| n | u | 128 |
| | | **1161** |

Figure 5: Breakdown: concordant/discordant assessments by the two raters.

Overall, the values in Table 5 show that both kappa and alpha coefficients are approx. 0.57, and both these values indicate a "moderate" agreement according to the magnitude scale for kappa (Sim and Wright, 2005), and the alpha range (Krippendorff, 2011). The moderate agreement between the two domain experts may suggest that selective experience and/or expertise could play a role in recognizing implant terms, and BERT terms can contribute in alerting professionals about the presence of implants that could otherwise be overlooked.

**Gold Standard and Term Clusters: *Intra-cluster Cleanliness*.** The evaluateD BERT terms are the first building block of a gold standard for this task. We use this "ground truth" to assess the quality of the individual term clusters. In this context, a term cluster is a group of words semantically-related to a glossary term used to build queries (see Section 4.1.3). Examples of term clusters are shown in the Appendix.

Since we will never know the number of True Negatives and False Negatives in this task, we cannot use traditional evaluation metrics. For this reason, we used a metric that we call "term cluster cleanliness" (short **cleanliness**) to roughly assess the linguistic quality and the term relatedness within a cluster.

Cleanliness is the proportion of True Positives (TP) with respect to the numbers of terms in the cluster, i.e.:

**Cleanliness= TP/(TP + FP + U + Disc + New)**

where:

**TP** (True Positives) is the number of terms that are classified as *indicative* of implants by both annotators in the gold standard.

**FP** (False Positives) is the number of terms that are classified as *non-indicative* of implants by both annotators in the gold standard;

**U** (Unsure) is the number of terms that both annotators agree on being unsure about whether they are indicative of implants or not;

**Disc** (Discordant) is the number of terms in the gold standard on which the annotators disagree upon.

**New** is the number of terms that are not in the gold standard but are in a cluster.

This metric is simple, but handy. Additionally, numbers can be easily swapped in the formula, so that it is possible to account for the proportion of new terms (Novelty) or Undecidedness, etc. For instance:

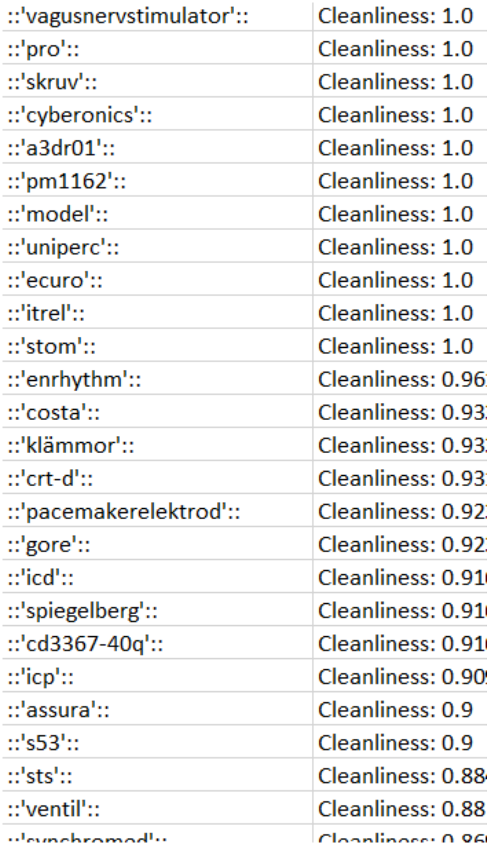
*Novelty* = New/(TP + FP + U + Disc + New)

*Undecidedness* = U/(TP + FP + U + Disc + New)

The cleanliness scores can be used to rank the term clusters and to set a threshold to trim out uninteresting terms (Figure 6 shows the top-ranked clusters returned by BallTree with extended glossary).

## 6 Discussion

The combination of searching the result space and the two versions of the glossary show that differing clusters are produced for the same glossary term (see the results for the glossary term 'ventil' in the Appendix, Figures A1, A2, A3 and A4). One possible way to unify these nuanced results would be to select the cluster with the highest cleanliness score for the same glossary term. For instance, for



| | |
|---|---|
| ::'vagusnervstimulator':: | Cleanliness: 1.0 |
| ::'pro':: | Cleanliness: 1.0 |
| ::'skruv':: | Cleanliness: 1.0 |
| ::'cyberonics':: | Cleanliness: 1.0 |
| ::'a3dr01':: | Cleanliness: 1.0 |
| ::'pm1162':: | Cleanliness: 1.0 |
| ::'model':: | Cleanliness: 1.0 |
| ::'uniperc':: | Cleanliness: 1.0 |
| ::'ecuro':: | Cleanliness: 1.0 |
| ::'itrel':: | Cleanliness: 1.0 |
| ::'stom':: | Cleanliness: 1.0 |
| ::'enrhythm':: | Cleanliness: 0.96 |
| ::'costa':: | Cleanliness: 0.93 |
| ::'klämmor':: | Cleanliness: 0.93 |
| ::'crt-d':: | Cleanliness: 0.93 |
| ::'pacemakerelektrod':: | Cleanliness: 0.92 |
| ::'gore':: | Cleanliness: 0.92 |
| ::'icd':: | Cleanliness: 0.91 |
| ::'spiegelberg':: | Cleanliness: 0.91 |
| ::'cd3367-40q':: | Cleanliness: 0.91 |
| ::'icp':: | Cleanliness: 0.90 |
| ::'assura':: | Cleanliness: 0.9 |
| ::'s53':: | Cleanliness: 0.9 |
| ::'sts':: | Cleanliness: 0.88 |
| ::'ventil':: | Cleanliness: 0.88 |
| ::'synchromed':: | Cleanliness: 0.86 |

Figure 6: Top-ranked term clusters (BallTree, extended glossary).

the term ‘ventil’, the best cluster is the one shown in Figure A2, since it has the best score.

Undeniably, the domain expertise is of fundamental importance for the refinement of the model, since the model sieve through extremely noisy textual data. The domain expert evaluation has helped us to identify the kind of irrelevant words the model retrieves. Error analysis indicates that families of irrelevant words negatively affect the quality of the clusters, e.g. named entities, like *Ann-Christin* (see Figure A5 in the Appendix) and general medical terms, like *aneurysm* (see Figure A6 in the Appendix). The next step is then to filter out semantic families of words that create noise in the results. However, this operation is not straightforward since there are some apparently non-indicative words (like ‘obs’, en: attention) that helped in the discovery of implant terms because they frequently co-occur with them (see Figure A7 in the Appendix). This means that ranking the term clusters based only on their cleanliness is helpful, but it does not tell the whole story about how indicative words can be in domain-specific contexts.

## 7 Conclusion

In this paper, we presented results of a BERT model for focused terminology extraction. The model was devised to discover terms indicative of implants in Swedish EMRs. Although the task is challenging, manual evaluation shows that the approach is rewarding, since a solid number of indicative terms were discovered by BERT. We used these BERT discoveries assessed by domain experts to create the first building block of a gold standard that we will use to evaluate future versions of our model. We plan the following:

- annotation of the “new” terms (cyan spheres in the figures in the Appendix) by the two rater; these terms and their annotation will be appended to the current gold standard;
- the removal of named entities mentioned in the texts of the EMRs;
- the removal of general medical terms;
- the cleansing of the noise in the glossaries using the non-indicative words annotated during the creation of the gold standard;
- the conflation of the cleaned baseline and extended glossary into a single one;
- a deeper understanding of the effect of fine-tuning parameters (e.g. the effect of a smaller block size);
- a more advanced search method of the result space to overcome the fragmentation of the corpus in data parts (chunks) of 50000 tokenized sentences at the time in order to avoid the re-merging of all the results at the end of this process.

## Acknowledgements

Thanks to the reviewers for insightful comments and useful suggestions. This research was funded by **Vinnova** (Sweden’s innovation agency), https://www.vinnova.se/ **Project title**: *Patient-Safe Magnetic Resonance Imaging Examination by AI-based Medical Screening*. **Call:** Start your AI-journey! For public organizations. **Grant number**: 2020-00228 to Peter Lundberg. The study was approved by the Swedish Ethical Review Authority (Etikprövningsmyndigheten), authorizaton nr.: 2021-00890 (to Peter Lundberg). ***Project Website***: http://www.santini.se/mri-terms/

## References

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

Kenneth Benoit, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, and Akitaka Matsuo. 2018. quanteda: An R package for the quantitative analysis of textual data. *Journal of Open Source Software*, 3(30):774.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Gabor Csardi and Tamas Nepusz. 2006. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695.

Hercules Dalianis, Martin Hassel, and Sumithra Velupillai. 2009. The Stockholm EPR corpus-characteristics and some initial findings. In *Proceedings of ISHIMR*, pages 243–249.

Leon Derczynski. 2020. Power consumption variation over activation functions. *arXiv preprint arXiv:2006.07237*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Aron Henriksson, Hans Moen, Maria Skeppstedt, Ann-Marie Eklund, Vidas Daudaravicius, and Martin Hassel. 2012. Synonym extraction of medical terms from clinical text using combinations of word space models. *Proceedings of Semantic Mining in Biomedicine (SMBM). Institute of Computational Linguistics, University of Zurich*, pages 10–17.

Ashwin Ittoo and Gosse Bouma. 2013. Term extraction from sparse, ungrammatical domain-specific documents. *Expert Systems with Applications*, 40(7):2530–2540.

Johan Kihlberg and Peter Lundberg. 2019. Improved workflow with implants gave more satisfied staff. In *SMRT 28th Annual Meeting 10-13 May 2019*.

Erik Kindberg. 2019. Word embeddings and patient records: The identification of MRI risk patients. B.sc. thesis, Linköping University.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv*, pages arXiv–1412.

Klaus Krippendorff. 1980. Content analysis. *California: Sage Publications*, 7:l–84.

Klaus Krippendorff. 2011. Computing Krippendorff's Alpha-Reliability. `http://repository.upenn.edu/asc_papers/43`.

Fei Li, Yonghao Jin, Weisong Liu, Bhanu Pratap Singh Rawat, Pengshan Cai, and Hong Yu. 2019. Fine-tuning bidirectional encoder representations from transformers (BERT)–based models on large-scale electronic health record notes: An empirical study. *JMIR medical informatics*, 7(3):e14830.

Tim Lustberg, Johan Van Soest, Peter Fick, Rianne Fijten, Tim Hendriks, Sander Puts, and Andre Dekker. 2018. Radiation oncology terminology linker: A step towards a linked data knowledge base. *Studies in health technology and informatics*, 247:855–859.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the national library of Sweden–making a Swedish BERT. *arXiv preprint arXiv:2007.01658*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Anton Nilsson, Jonathan Källbäcker, Julius Monsen, Linda Nilsson, Marianne Mattila, Martin Jakobsson, and Oskar Jerdhaf. 2020. Identifying implants in patient journals using BERT and glossary extraction. Student Report. Linköping University `http://www.santini.se/mri-terms/2020-06-04_ProjectReportGroup1-729G81_Final.pdf`.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8026–8037.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Viachaslau Sazonau, Uli Sattler, and Gavin Brown. 2015. General terminology induction in OWL. In *International Semantic Web Conference*, pages 533–550. Springer.

Elliot Schumacher and Mark Dredze. 2019. Learning unsupervised contextual representations for medical synonym discovery. *JAMIA open*, 2(4):538–546.

Julius Sim and Chris C Wright. 2005. The kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Physical therapy*, 85(3):257.

Darja Šmite, Claes Wohlin, Zane Galviņa, and Rafael Prikladnicki. 2014. An empirically based terminology and taxonomy for global software engineering. *Empirical Software Engineering*, 19(1):105–153.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

## Appendix

The graphs in this section are created with the R package Igraph[9] (Csardi and Nepusz, 2006).

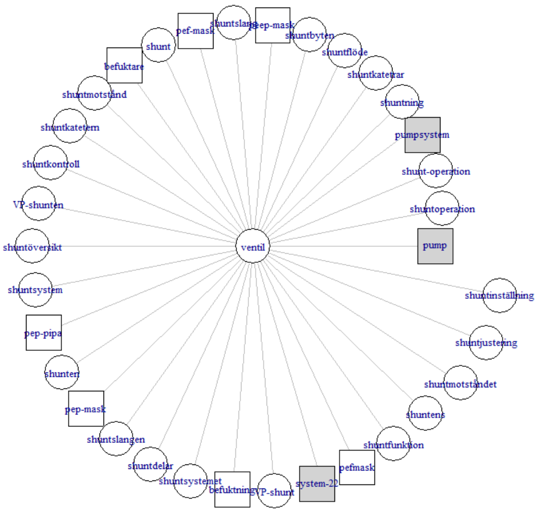


Figure A1: **KDTree, extended glossary: BERT terms related to 'ventil' (en: valve). Cleanliness: 0.70**
*Legend*: blank circles are TPs, blank squares are FPs, grey squares are Disc. The length of the edges represents the distance of a BERT term from the glossary term.

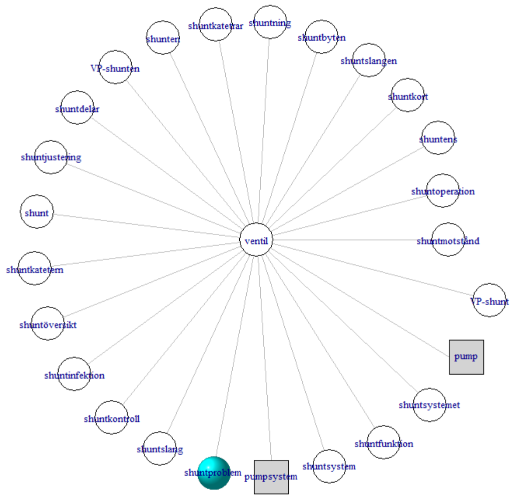


Figure A2: **BallTree, extended glossary: BERT terms related to 'ventil' (en: valve). Cleanliness: 0.88**
*Legend*: blank circles are TPs, blank squares are FPs, grey squares are Disc, cyan spheres are words that are not in the gold standard. The length of the edges represents the distance of a BERT term from the glossary term.

[9] https://cran.r-project.org/web/packages/igraph/index.html

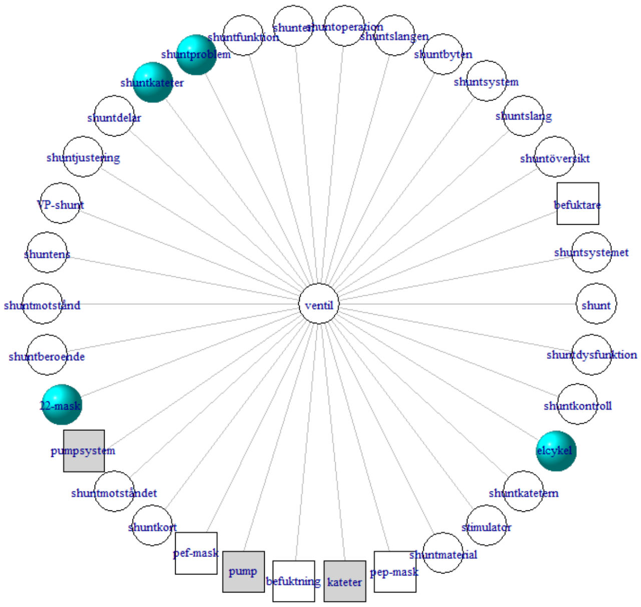


Figure A3: **KDTree, baseline glossary: BERT terms related to 'ventil' (en: valve). Cleanliness: 0.67**
*Legend*: blank circles are TPs, blank squares are FPs, grey squares are Disc, cyan spheres are words that are not in the gold standard. The length of the edges represents the distance of a BERT term from the glossary term.

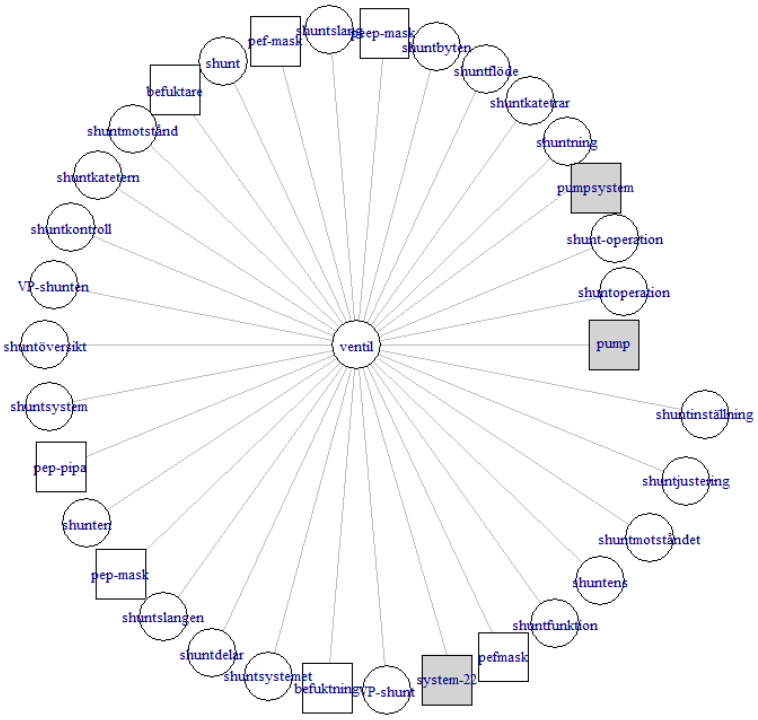


Figure A4: **BallTree, baseline glossary: BERT terms related to 'ventil' (en: valve). Cleanliness: 0.70**
*Legend*: blank circles are TPs, blank squares are FPs, grey squares are Disc. The length of the edges represents the distance of a BERT term from the glossary term.

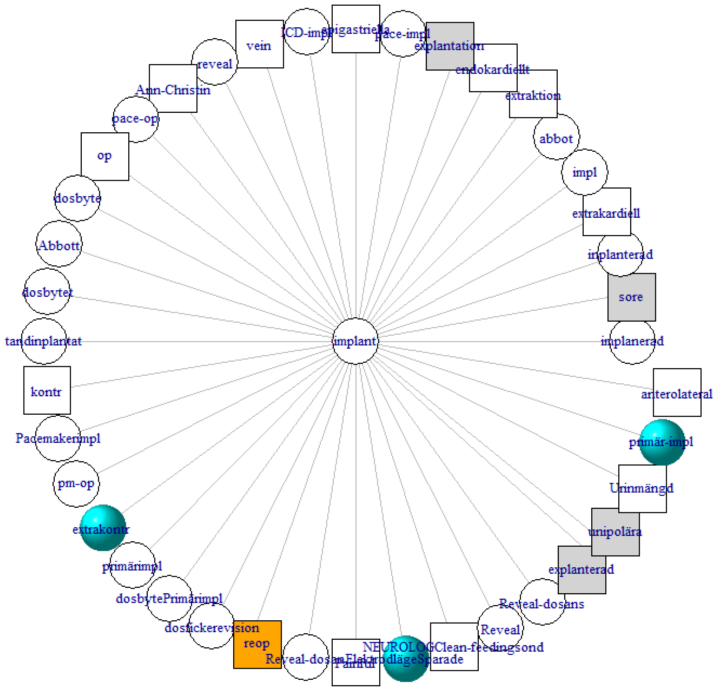


Figure A5: **BallTree, extended glossary: BERT terms related to 'implant' (in English in the glossary). Cleanliness: 0.5**
*Legend*: blank circles are TPs, blank squares are FPs, grey squares are Disc, orange squares are terms on which both raters are unsure about, cyan spheres are words that are not in the gold standard. The length of the edges represents the distance of a BERT term from the glossary term.

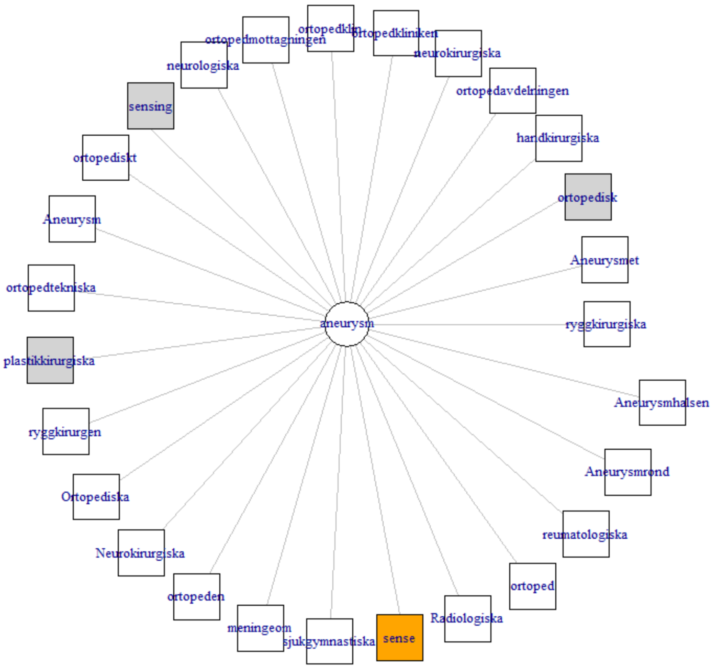


Figure A6: **BallTree, extended glossary: BERT terms related to 'aneurism'. Cleanliness: 0.0**
*Legend*: blank circles are TPs, blank squares are FPs, grey squares are Disc, orange squares are terms on which both raters are unsure about, cyan spheres are words that are not in the gold standard. The length of the edges represents the distance of a BERT term from the glossary term.

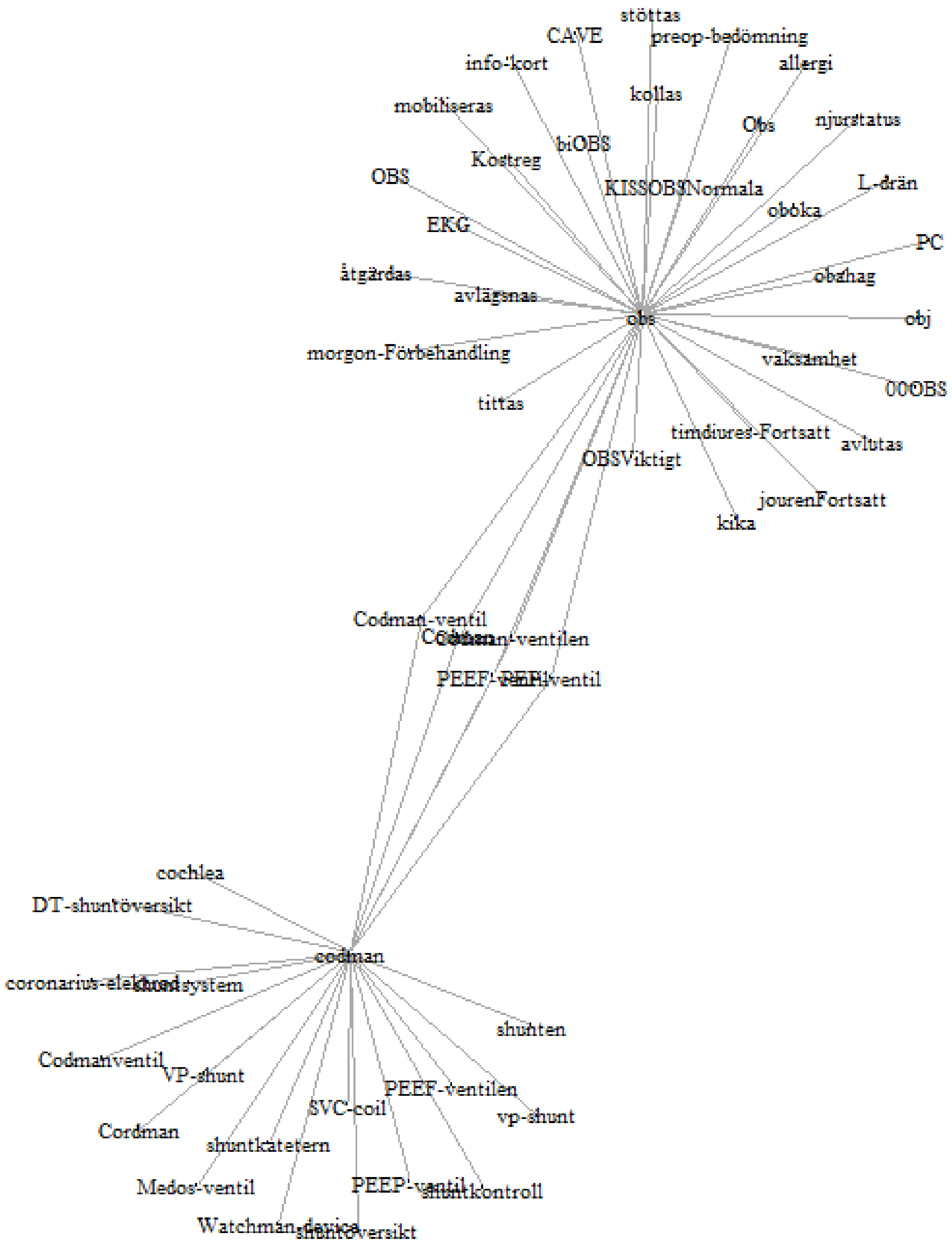


Figure A7: BallTree, extended glossary: relatedness between ‘codman’ and ‘obs

# Very necessary: the meaning of non-gradable modal adjectives in discourse contexts

**Maryam Rajestari**[1]
s8maraje@stud.uni-saarland.de

**Simon Dobnik**[2]
simon.dobnik@gu.se

**Robin Cooper**[2]
robin.cooper@ling.gu.se

**Aram Karimi**[2]
aram.karimi@gu.se

[1]Saarland University
Saarbrucken, Germany

[2]CLASP and FLOV
University of Gothenburg, Sweden

## Abstract

In this paper we provide a quantitative and qualitative analysis of meaning of allegedly non-gradable modal adjectives in different discourse contexts. The adjectives studied are *essential*, *necessary*, *crucial* and *vital* which are compared with a gradable modal adjective *important*. In our study sentences containing these adjectives were chosen from a large corpus together with their contexts. Then 120 English native speakers evaluated the meaning of these adjectives in a crowd-sourced study. Different types of contexts were chosen for this purpose. In some the adjectives were used as gradable with a modifier *very* while in others as non-gradable, without a modifier. We also modified the contexts by adding or removing the modifier *very*. The task for evaluators was to provide a replacement for adjectives for all the resulting contexts. From the replacements we are able to quantitatively evaluate the semantic potential of these contexts and what kind of adjectives they license.

## 1 Modality and adjectives

As a broad linguistic term, modality has newly gained increasing interest and been defined in different ways by linguists. (Huddleston and Pullum, 2002) argue that modality is mainly concerned with speakers' attitudes towards factuality or actualisation of the situation expressed by the rest of the clause. Comparing these two utterances "She wrote it herself" and "She must have written it", the first sentence, as a declarative main clause, is considered as non-modalised since no qualification or specialised emphasised has been made by the speaker towards the factuality of the preposition. By contrast, the second utterance is modalised since the truth of the preposition can only be indirectly inferred. By actualisation they refer to the utterances which have a relation to the future situation as in "She must help her friend." The two modalised utterances above ,however belong to two different kinds of modality, express "necessity" as the core idea. Other concepts can also be considered as main concepts besides "necessity" in the domain of modality, for instance, "possibility", "obligation" and "permission".

### 1.1 Linguistic Elements for Expressing Modality

(Matthewson, 2016) stated that languages vary in how they express and categorise modal meanings. For example in the Salish language St'át'imcets (Lillooet) the same morpheme (the enclitic =ka) can express either permission or obligation. A different morpheme (the circumfix ka- … -a) is used to express ability. In English, modality can be expressed by several parts of speech, for example, auxiliary verbs, verbs, adjectives, adverbs and nouns referred to as "Lexical Modals". Obviously there are other ways to express modality in English which are beyond the scope of the current paper. Referring to (Van Linden, 2012) who argues for a new definition for modal adjectives seemingly speakers or writers apply a kind of desirability scale to choose among adjectives from the same semantic domain in a specific situation. This desirability scale can also be applied when speakers or writers need to add weight to the chosen adjectives by using modifiers. The capability of adding a degree modifier to a modal adjective, is one criteria for the gradability evaluation. Gradability is expressed in the next section.

### 1.2 Degrees in Modal Adjectives

Among non-modal adjectives, a class has been known as the extreme adjectives, for example *big* has the extreme counterpart *huge* and *smart* has *brilliant* (Paradis, 2001) and (Rett, 2008). This distinction is also applicable to modal adjectives; for example, *crucial* and *certain* are extreme or strong modal adjectives, comparing with non-extreme or weak ones such as *important* and *likely*.

(Portner and Rubinstein, 2016) argue that strong modals cannot be gradable. Based on this we name extreme or strong modal adjectives as non-gradable modal adjectives and non-extreme or weak modal adjectives as gradable modal adjectives. The instances below taken from (Portner and Rubinstein, 2016) show the distinction between the two terms:

- Non-gradable modal adjectives:
  - A: It is crucial that our uninsured citizens get insurance.
  - B: And it's crucial that we allow people to make their own choices.
  - A: So we're stuck.
- Gradable modal adjectives:
  - A: It is important that our uninsured citizens get insurance.
  - B: It's also important that people make their own choices.
  - A: So how do we balance these things?

In the first example with the adjective *crucial*, Portner and Rubinstein point out that *A* and *B* are arguing that both of the following have the highest priority: *uninsured citizens must get insurance* and *we must allow people to make their own choices*. This leads to an impasse. In the second example with *important*, they argue, this impasse does not occur. However, there is a question whether the idea of non-gradability is as clear as this. This is because we find examples like the following:

> It is now widely apparent that the future of the earth as a living system is in many ways threatened, and that the basic cause is modern alienation from nature. There is a *very essential* difference between the present scientific way of regarding the earth, as a mass of inert matter, and the traditional view of it as a living, spiritual entity.

This suggests that modal adjectives are not straightforwardly distinguished as gradable or non-gradable. We argue that the gradability of modal adjectives is flexible and negotiable within the communicative context. Modifiers can coerce non-gradable modal adjectives to gradable ones. This view assumes that meaning of lexical items is not fixed but fluid, related to the contexts they are used in. It might be the case then that non-gradable modal adjectives have a potential to be coerced into gradable ones in different contexts (see, for example, Pustejovsky, 1995; Clark, 1996; Cooper and Kempson, 2008). Two research questions are considered in this study:

Q1 To what extent can "non-gradable" modal adjectives be used as gradable?

Q2 What is the meaning of non-gradable modal adjectives when they co-occur with degree modifiers?

To answer the first research question we perform a corpus study of examples of such usages in order to examine to what degree a modification of allegedly non-gradable adjectives is found in general language use or to what degree we should trust the linguists' intuitions cited in the previous work. For the second research question, we specifically examine how non-gradable modal adjectives behave when co-occurring with degree modifier "very" and how their meanings vary across different contexts.

The contributions of our study are both to theoretical linguistics and language technology. It investigates on the example of the corpus study to what degree structures that are traditionally left out from semantic analyses on the grounds that they do not exist occur in corpora of free text. We demonstrate that these are found in corpora and their semantics are captured by information theoretic measures. Knowing the semantic properties of these constructions gives important insights how such structures should be modelled and represented in feature-based annotation and rule-based approaches to language technology but also knowing what meaning representations we expect unsupervised language models to capture.

## 2 Q1: Gradable use?

We draw our examples of adjective use from the ukWaC dataset (Baroni et al., 2009). This is a large corpus of British English which contains more than a billion words ($N = 2,283,659,645$) sampled from websites in the UK domain. In order to answer the first research question, whether modifiers occur with "non-gradable" adjectives, we calculate log likelihood ratios as shown in Figure 1. With these we can test a hypothesis that a particular modifier and an adjective are collocated (h2) vs a hypothesis that the words are independent (h1). Firstly, looking at the co-occurrence counts for "very A" we see that in the ukWaC dataset we do find such examples. We also include "important" which is commonly agreed to be a gradable adjective. The statistical test in most cases confirms h2 that they are collocated (see column $p < 0.05$). In the last column we can see how many times the collocation hypothesis is more likely for that word combination than the hypothesis that the words are independent. The associations are very strong, e.g. 4.38e+7.

## 3 Q2: Meaning variation of gradable and non-gradable use

Our second research question addresses the semantics of allegedly non-gradable modal adjectives when they are used with and without a degree modifier. From the discussion in the previous section we have already rejected the possibility that all of them are non-gradable - why would they then occur with a degree modifier. Another possibility is that they are all gradable and there is no difference whether they are used with a degree modifier or not. A third possibility is that they can be gradable and non-gradable but gradability is contextually determined. Therefore, we would expect that modifiers will be associated with certain contexts more than others. To test these hypotheses, the following steps have been implemented.

From the ukWaC corpus we took sample sentences containing different "non-gradable" adjectives (*essential*, *crucial*, *necessary* and *vital*) as well as the gradable adjective *important* in their contexts. Each context consists of a target sentence containing one of the adjectives plus one preceding and one following sentence as follows: $S_{t-1}$ $S_t$ $S_{t+2}$ where $S_t$ is a target sentence. For example:

> "As soon as you can, you should arrange further supplies by contacting your GP surgery. It is *very vital* that you never run out of drugs. For information about each of the drugs named below, click on each link."

Two sets of 50 contexts were sampled: one set where in a target sentence an adjective co-occurred with the degree modifier "very" and the other set consisting of target sentences in which the adjectives did not occur with a degree modifier. From these another 50 contexts were created where the target sentences were modified by either removing or adding a degree modifier. The contexts are distributed as follows:

- 25 target sentences containing a modal adjective and a modifier (**very A**)
- 25 target sentences containing only a modal adjective (**A**)
- 25 modified target sentences (**~~very~~ A**) from the first set
- 25 modified target sentences (**+very A**) from the second set.

Participants were randomly assigned to one of the tasks. They were asked to provide the closest synonym for each adjective in the target sentence. This way, we can analyse the meaning variation of the provided synonyms in each context to confirm the hypothesis about context dependent meaning.

In particular, our hope is that the semantic similarity of synonyms within the context will be stronger than across the contexts. Equally, we are expecting more semantic similarity between synonyms in the original contexts than in the modified contexts.

The tasks were presented to 120 English native speakers in two ways: a crowd-sourcing task which we ran on the Semant-o-matic tool[1] and the Amazon Mechanical Turk (AMT). Semant-o-matic was designed for the purpose of online collection of linguistic data and can be targeted to particular informants. The AMT also allows us to collect a large number of judgements more quickly but the background of participants is less known: for example we can only restrict our task to domains of English speaking countries. To further check that our participants are native speakers we asked them, somewhat indirectly, to list languages that they speak, from best to worst. If a participant reported English as their first language we considered them a native speaker. The same interface was used in both data collection experiments.

The collected data was assessed for quality. We selected the high quality answers from AMT for our analysis by removing answers that were nonsensical. We removed all data from participants who provided more than 33% irrelevant answers.

Some of the instances following by the discussion are explained here.

## 4 Qualitative analysis

**Very vital and ~~very~~ vital** "vital" as a non-gradable adjective means "absolutely necessary", at the highest point in the scale of desirability. However, in the context below "vital" is used as gradable in the original context.

[1] http://www.dobnik.net/simon/semant-o-matic/

| Mod | A | C(Mod) | C(A) | C(Mod A) | $-2log\lambda$ | $p$ | $p < 0.05$ | $H_2$ vs. $H_1$ |
|---|---|---|---|---|---|---|---|---|
| very | necessary | 1990348 | 346547 | 740 | 450.95 | 4.47e-100 | 1 | 8.39e+97 |
| very | crucial | 1990348 | 69852 | 177 | 145.76 | 1.46e-33 | 1 | 4.49e+31 |
| very | vital | 1990348 | 115505 | 120 | 3.5 | 0.06 | 0 | 5.75 |
| very | essential | 1990348 | 225925 | 136 | 21.17 | 4.2e-6 | 1 | 3.96e+4 |
| very | compulsory | 1990348 | 41967 | 0 | 73.19 | 1.18e-17 | 1 | 7.80e+15 |
| very | certain | 1990348 | 314719 | 169 | 46.94 | 7.33e-12 | 1 | 1.56e+10 |
| very | important | 1990348 | 775926 | 41389 | inf | 0.0 | 0 | inf |
| very | appropriate | 1990348 | 403227 | 820 | 453.06 | 1.56e-100 | 1 | 2.40e+98 |
| very | proper | 1990348 | 107779 | 157 | 35.19 | 2.99e-9 | 1 | 4.38e+7 |
| very | likely | 1990348 | 365718 | 4989 | inf | 0.0 | 0 | inf |
| extremely | necessary | 147641 | 346547 | 21 | 0.09 | 0.76 | 0 | 1.05 |
| extremely | crucial | 147641 | 69852 | 15 | 15.05 | 1.05e-4 | 1 | 1.85e+3 |
| extremely | vital | 147641 | 115505 | 23 | 20.69 | 5.41e-6 | 1 | 3.10e+4 |
| extremely | essential | 147641 | 225925 | 11 | 0.97 | 0.32 | 0 | 1.63 |
| extremely | compulsory | 147641 | 41967 | 0 | 5.43 | 0.02 | 1 | 15.08 |
| extremely | certain | 147641 | 314719 | 2 | 27.42 | 1.64e-7 | 1 | 8.99e+5 |
| extremely | important | 147641 | 775926 | 5733 | inf | 0.0 | 0 | inf |
| extremely | appropriate | 147641 | 403227 | 20 | 1.54 | 0.21 | 0 | 2.16 |
| extremely | proper | 147641 | 107779 | 1 | 8.05 | 4.54e-3 | 1 | 56.09 |
| extremely | likely | 147641 | 365718 | 166 | 362.50 | 8.08e-82 | 1 | 5.22e+78 |
| fairly | necessary | 99431 | 346547 | 3 | 14.49 | 1.41e-4 | 1 | 1.40e+3 |
| fairly | crucial | 99431 | 69852 | 20 | 41.43 | 1.22e-10 | 1 | 9.90e+8 |
| fairly | vital | 99431 | 115505 | 7 | 0.69 | 0.41 | 0 | 1.41 |
| fairly | essential | 99431 | 225925 | 24 | 14.49 | 1.40e-4 | 1 | 1.40e+3 |
| fairly | compulsory | 99431 | 41967 | 0 | 3.65 | 0.06 | 0 | 6.22 |
| fairly | certain | 99431 | 314719 | 607 | inf | 0.0 | 1 | inf |
| fairly | important | 99431 | 775926 | 146 | 203.09 | 4.43e-46 | 1 | 1.26e+44 |
| fairly | appropriate | 99431 | 403227 | 4 | 15.28 | 9.26e-5 | 1 | 2.08e+3 |
| fairly | proper | 99431 | 107779 | 0 | 9.39 | 2.19e-3 | 1 | 109.17 |
| fairly | likely | 99431 | 365718 | 81 | 120.60 | 4.67e-28 | 1 | 1.54e+26 |

Figure 1: Gradable use of allegedly non-gradable adjectives and *important* with modifiers *very*, *extremely* and *fairly*. Similar results were also obtained for modifiers *really* and *absolutely*. $C$ are word counts; $\lambda$ is the log likelihood ration, $-2log\lambda$ os the log likelihood ratio approximated to the Chi-square statistic with a $p$ value. $H_2$ vs. $H_1$ tells us how many times $H_2$ is more likely than $H_1$. Values with inf cannot be reliably confirmed because they are too small to be calculated.

> “That’s the true value of literature and story – to give delight; and I’m very happy to see it given a home and a museum here in Oxford, where so many stories have begun.” Jacqueline Wilson, Children’s Laureate 2005-2007 ”Stories have always been a *very vital* part of my world, so a museum devoted to encouraging children to read and enjoy stories seems a wonderful idea. It’s especially fitting that it’s based in Oxford, which from Lewis Carroll onwards has always been associated with brilliant children’s literature.”

Figure 2 shows the synonyms provided by the annotators for both the original and modified contexts where the modifier *very* was removed. From the range of answers, we understand that “vital” in its gradable form can mean “important”, “necessary”, essential”, “central” and “consequential”. If we want to classify them, we may put all of these adjectives in the same range of meanings. However when the modified version of the context is considered (~~very~~ vital), other senses of meanings are also added to “vital”. When the local context of the adjective is modified by removing *very* there is a slight meaning shift in order to be able to fit into the remaining context of the sentence. This results in a larger number of possible synonym choices indicating a more dynamic interpretation of the adjective. Effectively, the modified sentences become more difficult to interpret and therefore the results become less congruent as individual participants are attempting different interpretations. It seems that this modified version forced the context to bear another sense of meaning such as “engaging”, “intrinsic”, “integral”, “chief”, “substantial”, “cornerstone”, “big” and “key”.

**Vital and +very vital** The variety of replacements was highly noticeable in the modified context where “vital” was used with a modifier. Figure 3 represents the variation clearly. Hence, this is in line to what we observed in the previous context which suggests that the gradability is linked to contexts. It appears that some contexts allow more or less gradability as seen for example in a slight difference in replacements for original contexts (very A vs A) between Figure 2 and 3.

| ~~very~~ vital | C | very vital | C |
|---|---|---|---|
| important | 6 | important | 11 |
| essential | 4 | essential | 3 |
| crucial | 3 | necessary | 3 |
| key | 3 | central | 1 |
| integral | 2 | consequential | 1 |
| intrinsic | 1 | | |
| engaging | 1 | | |
| chief | 1 | | |
| big | 1 | | |
| substantial | 1 | | |
| cornerstone | 1 | | |
| fundamental | 1 | | |

Figure 2: Answers obtained for "vital" in the original ("very vital") and modified contexts ("~~very~~ vital"). The results are ranked by counts (C).

| vital | C | +very vital | C |
|---|---|---|---|
| important | 3 | important | 5 |
| essential | 8 | essential | 1 |
| critical | 6 | critical | 6 |
| crucial | 7 | crucial | 7 |
| necessary | 2 | necessary | 3 |
| required | 1 | required | 1 |
| indispensable | 1 | imperative | 1 |
| principal | 1 | significant | 1 |
| | | integral | 1 |
| | | central | 1 |
| | | leading | 1 |

Figure 3: Answers obtained for "vital" in the original ("vital") and modified contexts ("+very vital").

> The pen/trap statute protects privacy and is an important investigative tool. Its application to the cyberworld is *vital*. Also, this legislation was passed in an era when telecommunication networks were configured in such a way that, in most cases, the information sought could be obtained by issuing an order to a single carrier.

**Necessary and +very necessary** Here is another context with the adjective *necessary*:

> "The bathroom is fully tiled and has a bath with overhead shower, bidet, w.c and wash hand basin. All the *necessary* bedding, bath and hand towels are provided. A useful store cupboard is located just inside the front door where the boiler is fitted."

In this special context where *necessary* was originally used without a modifier the replacement options are "needed", "required", "essential", and "requisite" as shown in Figure 4. Therefore, "necessary" here conveys a fixed range of meanings in the area of requirements. The degree of requirement can be determined from the context. The meaning-shift of the modified version is clearly observed. Having a gradable format of "necessary" instead of its non-gradable version in this context, leads to an increased ambiguity from a fixed range of meanings to a range of possible interpretations. Other senses were added by human evaluators.

| necessary | C | +very necessary | C |
|---|---|---|---|
| needed | 9 | important | 7 |
| required | 8 | essential | 6 |
| essential | 6 | needed | 3 |
| requisite | 2 | basic | 3 |
| fundamental | 1 | fundamental | 1 |
| important | 1 | required | 1 |
| indispensable | 1 | crucial | 1 |
| | | appropriate | 1 |
| | | vital | 1 |
| | | critical | 1 |

Figure 4: Answers obtained for "necessary" in the original ("necessary") and modified contexts ("+very necessary")

**Very necessary and ~~very~~ necessary** Here is an example of a context with *very necessary*:

> "It exists to further speleology and that means discovering, exploring and recording caves and other underground sites wherever they may be found. A *very necessary*, I would say essential, part of this is the recording. The club has two log books where members can write up their exploits and achievements."

As shown in Figure 5 in this specific context the adjective *necessary* has synonyms from "mandatory" to "important" with a limited number of other adjectives like "required", "needed", "essential". However, the meaning variation in the modified version in which "necessary" was used without a modifier is highly noticeable. Other meanings are added like "useful" and "basic". The degree to which adjectives can be replaced in the modified context seems to depend on the context itself, on the number of interpretations that can be reasonably constructed from it.

**Very crucial and ~~very~~ crucial** However, our experiment also shows that this distinction is not always so clear among the original and modified versions. Consider the following example:

| very necessary | C | ~~very~~ necessary | C |
|---|---|---|---|
| important | 8 | important | 2 |
| required | 1 | required | 4 |
| pivotal | 1 | significant | 2 |
| mandatory | 2 | mandatory | 1 |
| practical | 1 | main | 1 |
| crucial | 3 | crucial | 1 |
| vital | 1 | vital | 1 |
| needed | 2 | needed | 3 |
| critical | 4 | critical | 1 |
| essential | 1 | obligatory | 1 |
| fundamental | 2 | fundamental | 1 |
|  |  | basic | 1 |
|  |  | requisite | 1 |
|  |  | necessary | 1 |
|  |  | major | 1 |
|  |  | imperative | 1 |
|  |  | key | 2 |
|  |  | indispensable | 1 |
|  |  | incumbent | 1 |
|  |  | useful | 1 |

Figure 5: Answers obtained for “ necessary” in the original (“ very necessary”) and modified contexts (“~~very~~ necessary”)

> At beginning of course, when considering dialect, we looked at the relationship between social group identity and language. We considered the very crucial role that language plays in the formation and representation of identity. However, this account is limited in many senses.

As shown in Figure 6 in this specific context when *crucial* is used originally with a modifier the fluidity of meaning is observed to a higher degree than when it is used without a modifier.

In the next section we analyse this variation quantitatively using the measure of entropy which will give us a clearer picture to what extent this variation is possible in contexts and with adjectives chosen for this study.

## 5 Entropy as a measure of variation

To quantify the degree of variation of the replaced adjectives we calculate the entropy of their list $W$ for each ground truth adjective and context as follows:

$$H(W) = -\sum_{w \in W} p(w) log_2 p(w)$$

where $p$ is the likelihood of a word $w$ being used/replaced in a particular context by an AMT worker. Since different contexts result in different number of replacements we normalise the obtained entropies by the maximal attainable entropy which is $-log_2(n)$ where $n$ is the size of the set. If the normalised entropy of replaced synonyms is close to 1, it means that we are approaching maximum variation of answers and randomness (all items are equally probable) compared to when it approaches 0 and all the answers are the same and therefore completely predictable.

| very crucial | C | ~~very~~ crucial | C |
|---|---|---|---|
| important | 5 | important | 2 |
| essential | 2 | essential | 5 |
| critical | 5 | critical | 5 |
| vital | 3 | vital | 2 |
| significant | 1 | significant | 1 |
| paramount | 1 | paramount | 1 |
| central | 1 | central | 1 |
| serious | 1 | deciding | 1 |
| decisive | 1 | large | 1 |
| determining | 1 | key | 2 |
| necessary | 1 | all important | 1 |
| imperative | 1 | fundamental | 1 |
| prominent | 1 | pivotal | 1 |
| substantial | 1 | appropriate | 1 |
| big | 1 |  |  |
| mandatory | 1 |  |  |

Figure 6: Answers obtained for “ crucial” in the original (“ very crucial”) and modified contexts (“~~very~~ crucial”)

Figure 7 shows the meaning variation in very A/~~very~~ A compared to A/very A in our experiment. The red line shows the original contexts and the blue line shows the modified contexts. Adjectives under study are shown in the range of 5 in the horizontal lines which means 5 questions were devoted to each adjective. The vertical lines stand for the entropy result. Non-consistency among the adjectives can be inferred from the two figures which shows how adjectives behave differently with context consideration. It can be observed that how these adjectives mapped to the original and modified contexts sometimes with higher entropy and sometimes with lower entropy result. The detail of the entropy result is discussed in the next section.

### 5.1 Entropy Result over Original and Modified Contexts

As discussed in the previous section, the meaning of modal adjectives is fluid across different contexts. The entropy results support this idea of fluidity. Figure 8 shows the entropy results for the very A and ~~very~~ A condition. *important* as a commonly acceptable gradable modal adjective is also added for comparison. We can see that for *important* the difference in entropy of the answers for the original and the modified contexts is very small but for other adjectives “necessary”, “crucial”, “vi-

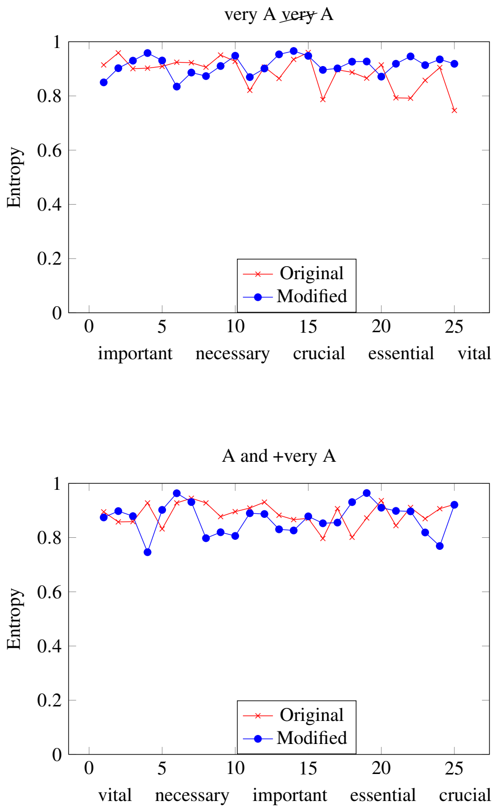


Figure 7: Normalised entropies over individual sentence contexts. The results indicate that overall individual examples have different variety of replacements for both comparisons. Modification ~~A~~ and +very A suggest different trends in entropy: this is further examined with t-tests in the text below.

tal” and “essential” it is larger. A two-tailed paired t-test found a significant difference between very-A versus ~~very~~-A (t(19)=2.179, p=0.042) for these adjectives (excluding important). Looking at individual adjectives more closely, it is likely that *necessary* is in alignment with *important* as they both get lower entropy of answers in the modified version which is not the case with *essential*, *crucial* and *vital*.

Next we compare the answers obtained from the A and +very A contexts shown in Figure 9. Excluding *important*, the two-tailed paired t-test found no significant difference between A versus +very-A (t(19)=1.003, p=0.3283) which is in contrast to the previous condition. In the second condition only *essential* got a lower average entropy in the original version.

### 5.2 Entropy Results across Contexts

We used the entropy analysis to compare the answers obtained from the original very A and A contexts as shown in Figure 10. The two-tailed paired t-test found no significant difference between very A versus A (t(19)=-0.4688, p=0.6445) for all adjectives excluding *important*.

Finally, the entropy analysis was done to compare the modified contexts (~~very~~ A and +A) as shown in Figure 11. A two-tailed paired t-test found a significant difference between +very A versus ~~very~~-A (t(19)=2.2808, p=0.0342) for all modal adjectives excluding *important*. This is expected since all of them are different contexts.

## 6 Discussion and Conclusion

Our findings, in particular, the log likelihood ratios, support the use of “non-gradable” adjectives with modifiers in a large corpus of British English. This demonstrates that the traditional distinction between gradable and non-gradable adjectives is not that straightforward.

However, what are the semantics of adjectives with a modifier and without a modifier is not straightforward when considering the analysis related to examination of synonym replacements. A possible explanation for our results is as follows. From the research on semantic coordination we know that the meaning of words shifts in contexts. Removing very (~~very~~ A) increases the entropy of synonyms while adding very (+very A) does not change the entropy of synonym replacements. Hence, if entropy of replacements corresponds to ambiguity, our explanation is that without a modifier an adjective is ambiguous between gradable and non-gradable reading, a form of underspecification. The interpretation is resolved within the context in which the adjective is used, including the communicative intent of the speaker. A context with very A will be non-gradable unambiguously by the virtue of the presence of the modifier and the non-gradable semantics must also be supported by the context, otherwise the utterance would not be well-formed. If we remove very, we therefore create a non-congruence with the non-gradable context since now also a non-gradable interpretation is at play. This leads to an increase in ambiguity of the sentence and an increase in entropy. On the other hand, original contexts without a modifier are ambiguous between gradable and non-gradable interpretations. Adding very simply selects a pref-

| Adjectives | very A | stdev | Rank | ~~very~~ A | stdev | Rank | diff |
|---|---|---|---|---|---|---|---|
| important | 0.9171 | 0.024 | 4 | 0.9143 | 0.041 | 3 | -0.003 |
| necessary | 0.9263 | 0.016 | 5 | 0.8906 | 0.0424 | 1 | -0.036 |
| crucial | 0.8974 | 0.0557 | 3 | 0.9276 | 0.0407 | 5 | 0.03 |
| essential | 0.8701 | 0.0498 | 2 | 0.9045 | 0.0233 | 2 | 0.034 |
| vital | 0.8188 | 0.0623 | 1 | 0.9263 | 0.0135 | 4 | 0.108 |

Figure 8: Average entropies over contexts for very A and ~~very~~ A. The ~~very~~ A leads to a lower entropy except for *important* and *necessary*.

| Adjectives | A | stdev | rank | +very A | stdev | rank | diff |
|---|---|---|---|---|---|---|---|
| important | 0.8918 | 0.0274 | 4 | 0.8624 | 0.0314 | 3 | -0.029 |
| necessary | 0.9148 | 0.0275 | 5 | 0.0776 | 0.8636 | 4 | -0.051 |
| crucial | 0.8908 | 0.0328 | 3 | 0.8608 | 0.0643 | 2 | -0.03 |
| essential | 0.8626 | 0.0625 | 1 | 0.9027 | 0.0484 | 5 | 0.04 |
| vital | 0.8743 | 0.0374 | 2 | 0.8599 | 0.0647 | 1 | -0.014 |

Figure 9: Average entropies over contexts for A and +very A. A leads to a higher entropy except for *essential*.

| Adjectives | very A | stdev | Rank | A | stdev | Rank | diff |
|---|---|---|---|---|---|---|---|
| important | 0.9171 | 0.024 | 4 | 0.8918 | 0.0274 | 4 | - 0.043 |
| necessary | 0.9263 | 0.016 | 5 | 0.9148 | 0.0275 | 5 | - 0.012 |
| crucial | 0.8974 | 0.0557 | 3 | 0.8908 | 0.0328 | 3 | - 0.006 |
| essential | 0.8701 | 0.0498 | 2 | 0.8626 | 0.0625 | 1 | - 0.007 |
| vital | 0.8188 | 0.0623 | 1 | 0.8743 | 0.0374 | 2 | 0.072 |

Figure 10: Average entropies and ranking over original contexts for very A and A.

| Adjectives | +very A | stdv | Rank | ~~very~~ A | stdv | Rank | diff |
|---|---|---|---|---|---|---|---|
| important | 0.8624 | 0.0314 | 3 | 0.9143 | 0.041 | 3 | -0.052 |
| necessary | 0.8636 | 0.0776 | 4 | 0.8906 | 0.0424 | 1 | -0.027 |
| crucial | 0.8608 | 0.0643 | 2 | 0.9276 | 0.0407 | 5 | -0.067 |
| essential | 0.9027 | 0.0484 | 5 | 0.9045 | 0.0233 | 2 | -0.002 |
| vital | 0.8599 | 0.0647 | 1 | 0.9263 | 0.0135 | 4 | -0.066 |

Figure 11: Average entropies and ranking over modified contexts for +very A and ~~very~~ A.

erence for a non-gradable interpretation which was already available and therefore there is no change in entropy. Natural contexts very A and A have identical entropy distribution; while changed contexts ~~very~~ A and +very A have different entropy distributions which means that they are affected differently by the modification. This provides further support for the hypothesis that modification is linked to a loss of congruence with the context (~~very~~ A) and therefore increase in ambiguity or resolution of ambiguity (+very A) towards non-gradable use.

With the analysis of synonym replacements with the information theoretic measure of entropy we tried to evaluate what is the semantic potential of the context with the adjectives and a potential gradable modifier *very*. We have linked the variation to the ambiguity of the contexts: the higher the ambiguity of a context the higher potential for using adjectives in this context. In our future work we intend to compare the potential replacements at a more fine-grained level by comparing their contextual word embeddings (Devlin et al., 2018) with the word embeddings of the original adjective. We hope that the exercise will also contribute to the evaluation of contextual word-embeddings as the task that we are interested is a highly fine-grained semantic task that tries to evaluate semantic differences *within* a particular class of part of speech.

## 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*, 43(3s):209–226.

Herbert H. Clark. 1996. *Using language*. Cambridge University Press, Cambridge.

Robin Cooper and Ruth Kempson, editors. 2008. *Language in Flux: Dialogue Coordination, Language Variation, Change and Evolution*, volume 1 of *Communication, Mind and Language*. College Publications, London.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv*, arXiv:1810.04805 [cs.CL]:1–14.

R Huddleston and G Pullum. 2002. *The Cambridge Grammar of English Language*. Cambridge University Press.

L Matthewson. 2016. Modality. In Maria Aloni and Paul Dekker, editors, *The Cambridge Handbook of Formal Semantics*, chapter 18, pages 525–559. Cambridge University Press.

Carita Paradis. 2001. Adjectives and boundedness. *Cognitive Linguistics*, 12(1):47–65.

Paul Portner and Aynat Rubinstein. 2016. Extreme and non-extreme deontic modals. In Nate Charlow and Matthew Chrisman, editors, *Deontic modality*, chapter 9, pages 256–282. Oxford University Press.

James Pustejovsky. 1995. *The Generative Lexicon*. MIT Press, Cambridge, Mass.

Jessica Rett. 2008. *Degree Modification in Natural Language*. Ph.D. thesis, Rutgers University Graduate School, New Brunswick.

An Van Linden. 2012. *Modal Adjectives: English Deontic and Evaluative Constructions in Synchrony and Diachrony*. Hubert & Co. GmbH & Co. KG, Göttingen, Printed in Germany.

# Granska API – an Online API for Grammar Checking and Other NLP Services

**Jonas Sjöbergh**
Theoretical Computer Science, KTH
Stockholm, Sweden
jsh@kth.se

**Viggo Kann**
Theoretical Computer Science, KTH
Stockholm, Sweden
viggo@nada.kth.se

## Abstract

We present an online API to access a number of Natural Language Processing services developed at KTH[1]. The services work on Swedish text. They include tokenization, part-of-speech tagging, shallow parsing, compound word analysis, word inflection, lemmatization, spelling error detection and correction, grammar checking, and more. The services can be accessed in several ways, including a RESTful interface, direct socket communication, and premade Web forms. The services are open to anyone. The source code is also freely available making it possible to set up another server or run the tools locally. We have also evaluated the performance of several of the services and compared them to other available systems. Both the precision and the recall for the Granska grammar checker are higher than for both *Microsoft Word* and *Google Docs*. The evaluation also shows that the recall is greatly improved when combining all the grammar checking services in the API, compared to any one method, and combining services is made easy by the API.

## 1 Introduction

A number of Natural Language Processing (NLP) tools for analysis of Swedish text have been developed at KTH (Kann, 2010). Most of the tools were developed in projects focused on grammar checking: "Algoritmer för svenska språkverktyg" ("Algorithms for Swedish language tools"), "Svensk grammatikgranskning" ("Swedish grammar checking"), and "CrossCheck – svensk grammatikkontroll för andraspråksskribenter" ("CrossCheck – Swedish grammar checking for second language writers"). This lead to a focus on tools that are useful for grammar checking, but low level language analysis tools useful in other applications are also included.

The source code for the tools has been freely available since the tools were developed, and anyone is free to install and use them locally. Now we have also made an online API available[2]. It can be used to access the tools running as services on a server at KTH, and these services are also open for anyone to use. They can be used by a user by hand, typing text or copying text from some other program, and by programs using the services to do some analysis they need.

We have also built an example application that uses the services to create a graphical text exploration environment, and we have evaluated some of the tools provided in the API, comparing them to other available systems that perform the same service.

## 2 Available Services

The available services can be divided into three types of services: low-level or preprocessing NLP tools that can be used to build more advanced services, tools to help when developing and evaluating NLP tools, and high-level NLP services that are directly useful to end users. It is still possible to build new tools on top of the high-level services.

The low-level services in the Granska API are:

**Tokenization** The Granska tokenizer tokenizes text into words and sentences. It is integrated in the Granska tagger and in the Granska grammar checker below, but it is also possible to build a stand-alone tokenizer.

**PoS Tagging** The Granska tagger (Carlberger and Kann, 1999) does part-of-speech tagging of Swedish text. It is a Hidden Markov Model tagger trained on the SUC corpus (Ejerhed et al., 1992) using a slightly modified version of the SUC tag set. The tagger is integrated

[1] A short version of this paper was presented at the SLTC-2020 conference.

[2] https://skrutten.csc.kth.se/granskaapi/

in the Granska grammar checker but can also run as a stand-alone application.

**PoS Tagging without context** Taggstava (Kann, 2010) is a tagger that assigns part-of-speech tags to words without using context information. It uses inflection rules for Swedish to determine what inflected form a word could be. No disambiguation is done for ambiguous words, all possible tags are returned. Taggstava uses the same rules and reference data as the spelling error detection program Stava below.

**Shallow Parsing** The GTA parser (Knutsson et al., 2003) does shallow parsing of Swedish text based on hand written rules. It identifies clause boundaries and phrases. The internal structures of phrases are identified, e.g. a noun phrase being part of a prepositional phrase, but a full tree for the whole sentence is not built.

GTA is built to be robust to noisy data (i.e. text with many errors) since it is built for and used in the grammar checker Granska below, which is expected to run on texts with possibly very many errors in them.

For convenience, there are also two services that return subsets of the GTA information, one that returns only clause boundaries, and one that returns only the phrase structure.

**Compound Word Analysis** SärStava (Sjöbergh and Kann, 2004) is a tool that gives the most likely interpretation of a compound word, or all possible interpretations. Possible interpretations are found using the Stava compound word analysis methods. Then statistical data and some heuristics are used to decide which interpretation is most likely for ambiguous compounds. No methods using the context of the word are used, though.

**Word Inflection** The Granska Inflector inflects Swedish words. It can generate a specific inflected form or a list of all possible inflections.

**Lemmatization** This service uses the Granska tagger to find the lemma form of words.

**Word-Tag-Lemma** Several other services expect the input to be triples of word, part-of-speech tag, and lemma form of the word. For convenience, a service that takes plain text and provides word-tag-lemma triples, by calling the Granska tagger, is also provided.

There is currently only one service in the development and evaluation tools category. Other tools for evaluating NLP tools are available to run locally, but have not been made available as online services yet. The available tool is:

**Realistic Spelling Error Generation** Missplel (Bigert et al., 2003) is a tool that automatically inserts spelling errors in texts. Different types of errors can be simulated, for example keyboard mistypes where a neighboring key is pressed by mistake or sound-alike errors where the writer may not know the correct spelling of a word they know how to say.

Missplel can be used to automatically evaluate the robustness of other NLP systems by showing how the performance degrades when there are errors in the text. For example, an evaluation can be done by running a parser on a test text and then running it on the same text with added errors. Ideally, the parser should produce similar output the second time, since the "intended" meaning of the text is the same. This way the robustness can be evaluated without using any annotated data.

The high-level services are all spelling and grammar checking services, since the tools were built in research projects focused on this. The available services are:

**Spelling Error Detection and Correction** Stava (Domeij et al., 1994) is a very powerful spelling correction tool for Swedish that finds spelling errors and suggests corrections. Stava handles the very productive compounding in Swedish using rules for how compounds can and cannot be created in Swedish. The compound analysis can also be accessed separately, as mentioned above. In Swedish it is very common to create new compound words in normal text, and without some form of compounding analysis there are normally very many false alarms from spelling error detection tools.

**Grammar Checking using Rules** The Granska (Domeij et al., 2000) system detects grammatical errors in Swedish text based on manually

Goal: get the most likely interpretation of the compound “glasstrut”.
API call: `https://skrutten.csc.kth.se/granskaapi/compound/best/glasstrut`
Output: `glasstrut glass|strut`

Goal: get all possible interpretations of the compound “glasstrut”, in JSON.
API call: `https://skrutten.csc.kth.se/granskaapi/compound/json/all/glasstrut`
Output: `["word":"glasstrut", "parts":["glas|strut", "glass|strut", "glass|trut"]]`

Goal: get phrase structure in the sentence “GTA kan analysera svensk text.”.
API call: `http://skrutten.csc.kth.se/granskaapi/chunk?text=GTA+kan+analysera+svensk+text+.`
Output: `GTA NPB, kan VCB, analysera VCI, svensk APMINB|NPB, text NPI, .  0`

Figure 1: Example API calls and the corresponding outputs

written error detection rules. The rule language (Knutsson et al., 2001) is quite powerful and the rule writer has access to all the information provided by the tools mentioned above. Rules can for example be written to allow suspicious things if they cross a phrase boundary (to reduce false alarms), or to change the inflected form of a suspicious word to a form more suitable to the surrounding context using the inflector above, etc.

Extra rules can be added to each API call. These can be used to detect new types of errors not covered by the standard rules or to influence the behavior of Granska (e.g. by adding more parsing rules). Here is an example of a simple rule:

```
altcorr@kong{
 X(wordcl=dt),
 Y(wordcl=nn & num!=X.num)
  -->
 corr(X.form(num:=Y.num))
 corr(Y.form(num:=X.num))
 action(scrutinizing)}
```

This rule finds places where a determiner (word class is “dt”) is followed by a noun (word class “nn”), but they have different number, i.e. it finds agreement errors since determiners and nouns should normally have the same number in Swedish. It then suggests two possible corrections, changing the number of the determiner or changing the number of the noun. The suggested corrections are generated with the inflector above.

**Grammar Checking using PoS n-grams**

ProbCheck (Bigert and Knutsson, 2002) detects grammatical errors in text using statistical analysis of part-of-speech n-grams. Based on n-gram statistics from correct text, it finds part-of-speech sequences that are rare in the reference data. It also uses the GTA parser above, since phrase and clause boundaries can cause very rare PoS n-grams even in correct text and thus lead to false alarms. ProbCheck usually runs integrated in Granska but running only ProbCheck is also possible.

ProbCheck was created in a project focused on helping second language learners. Learners of a language make many unpredictable errors that it can be hard to write error detection rules for. There are also generally a lot of errors, and thus not much correct text as context to base error detection rules on.

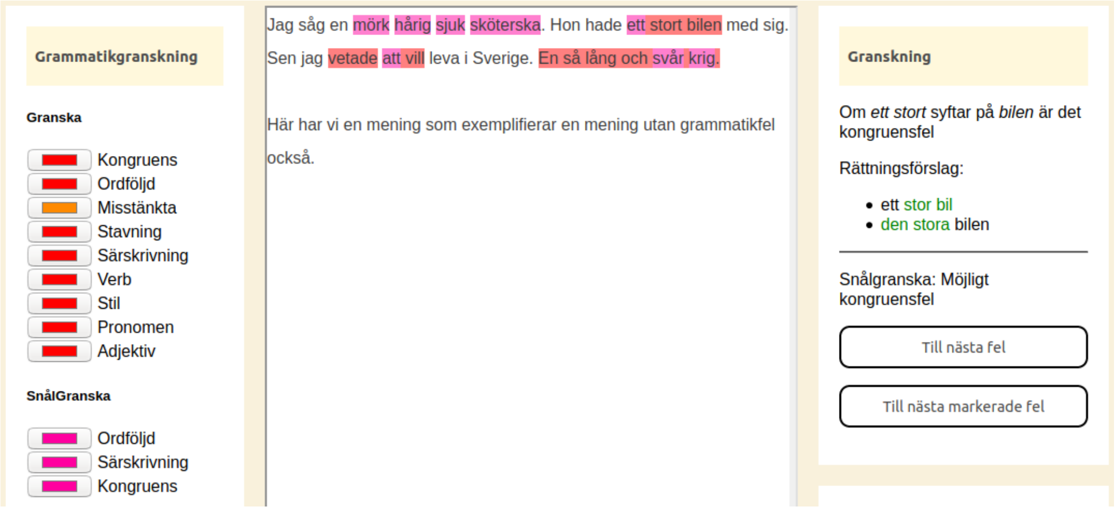
**Grammar Checking using Machine Learning**

SnålGranska (Sjöbergh and Knutsson, 2005) detects grammatical errors using machine learning trained on texts with synthetic errors added. By itself it does not perform as well as Granska, but it does detect errors that Granska does not detect, and it is possible to use both systems together to get improved coverage (Bigert et al., 2004).

## 3 Ways to Access the Services

All the services mentioned in the previous section can be accessed online. There are simple Web forms where you can enter words or text by hand (or by copy-paste from other applications) and see what the tools can do.

There is also a RESTful API to access the services. This allows typing in requests in the URL bar of a Web browser by hand, but is mainly intended for other programs to automatically use the services for language processing tasks they may need. Most



Jag såg en mörk hårig sjuk sköterska. Hon hade ett stort bilen med sig. Sen jag vetade att vill leva i Sverige. En så lång och svår krig.

Här har vi en mening som exemplifierar en mening utan grammatikfel också.

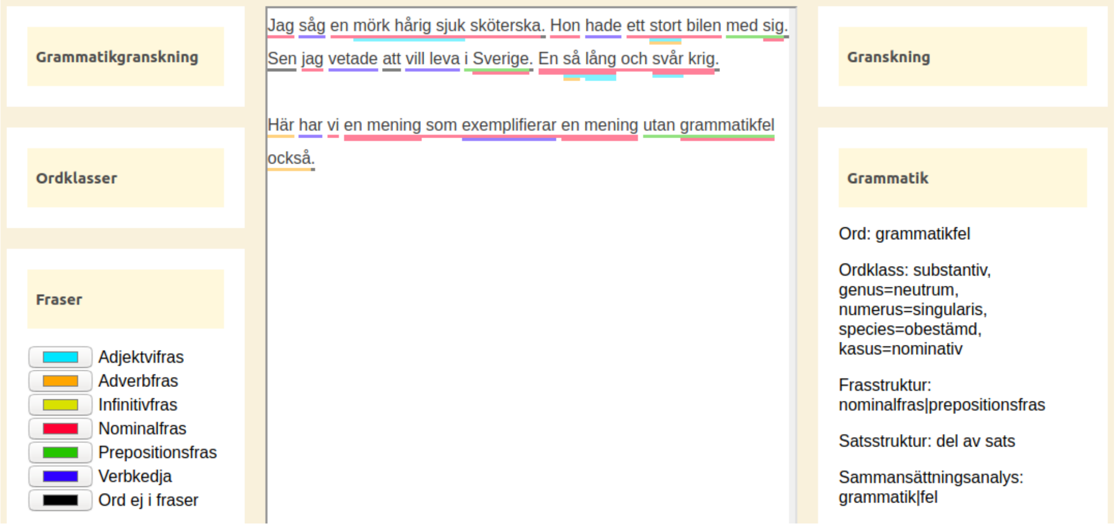
Om *ett stort* syftar på *bilen* är det kongruensfel

Rättningsförslag:

- ett stor bil
- den stora bilen

Snålgranska: Möjligt kongruensfel

Figure 2: FörHandsGranska, built on top of the API services. Here working as a text editor with spelling and grammar checking support, letting the user use suggested corrections through simple clicks.



Jag såg en mörk hårig sjuk sköterska. Hon hade ett stort bilen med sig. Sen jag vetade att vill leva i Sverige. En så lång och svår krig.

Här har vi en mening som exemplifierar en mening utan grammatikfel också.

Ord: grammatikfel

Ordklass: substantiv, genus=neutrum, numerus=singularis, species=obestämd, kasus=nominativ

Frasstruktur: nominalfras|prepositionsfras

Satsstruktur: del av sats

Sammansättningsanalys: grammatik|fel

Figure 3: FörHandsGranska, showing linguistic analysis. Words can be colored based on part-of-speech, phrase structure, or clause structure. Compound word analysis, possible inflections, etc., are also shown.

services can send back the reply in either plain text form, HTML, JSON, or XML. Figure 1 shows example API calls and the corresponding outputs.

It is also possible to access the services using socket communication. When communicating with the services directly using a socket, most services will only return the raw output of the original tool (for example not provide the result in JSON).

If no input is given, each service will display a Web page with information on how to call the service. An example Web form that uses the service is shown, and this can be used as a reference to see what input is expected, how to format the input, etc. A few example words or sentences are also provided to give a quick overview of what typical input and output can be expected to look like.

The API allows building new tools based on the services, creating new interfaces to the services, or integrating the services into existing tools (e.g. an editor or word processor). If an online service is not suitable, for example for a system that is expected to run offline, the source code for all the tools is also freely available. This makes it possible to install any tool and run it locally, or to install tools and set up a new server that can provide the same services.

## 4 Example Application based on the Services

We have created an example application using a number of the services described above.

| System | True Pos. | Diag. Errors | False Pos. | Total Reports | False Neg. | Precision (%) | Pseudo Recall (%) |
|---|---|---|---|---|---|---|---|
| Granska | 211 | 4 | 203 | 414 | 297 | 51 | 42 |
| ProbCheck | 130 | - | 468 | 598 | 378 | 22 | 26 |
| SnålGranska | 112 | 103 | 736 | 848 | 396 | 13 | 22 |
| All Granska API | 421 | 86 | 1262 | 1683 | 87 | 25 | 83 |
| MS Word | 64 | 2 | 270 | 334 | 444 | 19 | 13 |
| Google Docs | 107 | 5 | 663 | 770 | 401 | 14 | 21 |

Table 1: Evaluation on text with few errors (published novels). 508 errors annotated in 101,279 tokens. Pseudo recall (and False Negatives) is calculated based on all the errors found by any system, but since there are also errors not found by any system the true recall is lower. "Diag. Errors" are error reports where there is an error in the text but the diagnosis is wrong, for example reporting an agreement error when it is actually a spelling error. ProbCheck does not give error diagnoses.

| System | True Pos. | Diag. Errors | False Pos. | Total Reports | False Neg. | Precision (%) | Pseudo Recall (%) |
|---|---|---|---|---|---|---|---|
| Granska | 978 | 85 | 581 | 1559 | 888 | 63 | 52 |
| ProbCheck | 341 | - | 507 | 848 | 1525 | 40 | 18 |
| SnålGranska | 497 | 428 | 763 | 1260 | 1369 | 39 | 27 |
| All Granska API | 1579 | 374 | 1694 | 3273 | 287 | 48 | 85 |
| MS Word | 562 | 50 | 376 | 938 | 1304 | 60 | 30 |
| Google Docs | 360 | 8 | 407 | 767 | 1506 | 47 | 19 |

Table 2: Evaluation on blog texts, 1,866 errors annotated in 97,645 tokens. Pseudo recall (and False Negatives) is calculated based on all the errors found by any system, but since there are also errors not found by any system the true recall is lower. "Diag. Errors" are error reports where there is an error in the text but the diagnosis is wrong, for example reporting an agreement error when it is actually a spelling error. ProbCheck does not give error diagnoses.

FörHandsGranska[3] is a graphical text exploration tool. It can mark writing errors in different colors and suggest corrections, working as an editor with built in spelling and grammar checking tools. Errors in the text can be replaced with corrected text by simply clicking on suggestions from the grammar checking tools.

It can also add linguistic markup, coloring words based on their part-of-speech, underlining different types of phrases in different colors, or show clause boundaries. It also shows all inflections of a word, the compound analysis of compound words, and more. In this way, it can be used as a linguistic exploration tool or language learning tool. Interaction is also possible through for example clicking on a listed inflected form to change the inflection of the word in the original text.

It is possible to show both suspected writing errors and linguistic markup at the same time. Adding more rules in the Granska rule language is also supported. Two example screenshots of FörHandsGranska are shown in Figures 2 and 3.

FörHandsGranska also allows quick lookup in other online services not provided by the Granska API, such as the *SAOB* dictionary, the *Lexin* search service, or concordance lookup in the *Korp* service.

FörHandsGranska is written in JavaScript and is basically a graphical interface that calls the services of the Granska API when language analysis is needed.

## 5 Evaluation

There have been many evaluations of the different tools. For evaluations of the individual tools, we refer to the respective publications cited above.

We have also done a new evaluation using the tools through the Granska API. We have evaluated the Granska grammar checking tool on Swedish text. Since Granska also uses almost all of the other tools in the API, this gives an overview of how well all the tools can work together.

We fed unannotated Swedish text to the Granska API. For comparison, we also fed the same text

[3] https://skrutten.csc.kth.se/fhg/

to the spelling and grammar checking tool integrated in *Microsoft Word 2016*, the spelling and grammar checking tool in *Google Docs*, and the grammar checkers ProbCheck and SnålGranska. We also combined all the grammar checking services in the Granska API (Granska, ProbCheck, and SnålGranska) as one grammar checking service to see how much the recall improves by using several methods that hopefully complement each other.

All error reports from the grammar checkers were manually annotated as correct or not, but we did not manually check the text for errors not found by any of the grammar checking tools. A quick manual check of a small sample of the evaluation data showed that there are indeed errors that are missed by all the grammar checking tools.

The evaluation texts used all come from the *Språkbanken* corpus resources[4]. There are many corpora available for download, and there is a search interface with NLP tools that can be used to search the available corpora (Borin et al., 2012).

Table 1 shows the evaluation results on texts with few errors to find, in this case texts from published novels. Since there are few true errors to be found, precision can be expected to be low.

Table 2 shows the evaluation results on blog texts, which have more errors than the published novels. As expected, all grammar checking methods achieve higher precision in this test set.

The results support the idea that the different grammar checkers complement each other (as mentioned in Section 2, ProbCheck was explicitly created to complement Granska) since no grammar checker found even half of the total errors in the published novels and only one system found just over half the errors in the blog texts, when compared to all errors found by all the systems in total.

Using the Granska API it is easy to combine the output from any system included in the API. Combining the three services provided in the Granska API gives much higher recall than any single system achieves, as seen in Tables 1 and 2, though the precision is of course lower than the precision of the highest performing individual system.

The results also indicate that the grammar checking methods in the Granska API perform competitively when compared to other grammar checking systems for Swedish. Both the precision and the recall of the Granska grammar checker is higher than those of the grammar checking methods in both *Microsoft Word* and *Google Docs* in these test sets.

## 6 Related Work

There are other NLP APIs, both online APIs and APIs for using tools locally. Most APIs are for English but APIs for other languages are also available.

For Swedish, the *Sparv* corpus annotation pipeline (Borin et al., 2016) has an online API. It supports tokenization, lemmatization, part-of-speech tagging, compound analysis, dependency parsing, named entity recognition, and more. *Sparv* also supports languages other than Swedish.

The *SVENSK* project (Gambäck and Olsson, 2000) collected NLP tools for Swedish, including part-of-speech tagging, parsing, text classification, and more. Resources from different sources were integrated into one consistent framework using *GATE* (Cunningham et al., 1996).

## 7 Conclusions

We provide an online API to access NLP services for Swedish text. Both low level services like part-of-speech tagging and high level services like grammar checking are provided. The services are freely available online, with several ways to access them. The source code is also freely available, allowing users to set up their own servers or run the tools locally. The tools can be used by hand or integrated in other programs. As an example of what can be done by using the API, we have also created an online application for interactive text exploration that uses the API for all linguistic analysis needed.

We evaluated some of the higher level services, that in turn use most of the low level services, comparing them to other available systems. The results show that the performance is improved by combining several services, and that the provided services in themselves perform competitively compared to other available systems. Combining systems is easy using the provided API.

## References

Johnny Bigert, Linus Ericson, and Antoine Solis. 2003. Missplel and AutoEval: Two generic tools for automatic evaluation. In *Proceedings of Nodalida 2003*, Reykjavik, Iceland.

Johnny Bigert, Viggo Kann, Ola Knutsson, and Jonas Sjöbergh. 2004. Grammar checking for Swedish

[4] https://spraakbanken.gu.se/en/resources

second language learners. In Peter Juel Henrichsen, editor, *CALL for the Nordic Languages*, pages 33–47. Samfundslitteratur.

Johnny Bigert and Ola Knutsson. 2002. Robust error detection: A hybrid approach combining unsupervised error detection and linguistic knowledge. In *Proceedings of Romand 2002, Robust Methods in Analysis of Natural Language Data*, pages 10–19, Frascati, Italy.

Lars Borin, Markus Forsberg, Martin Hammarstedt, Dan Rosén, Roland Schäfer, and Anne Schumacher. 2016. Sparv: Språkbanken's corpus annotation pipeline infrastructure. In *Proceedings of SLTC 2016*, Umeå, Sweden.

Lars Borin, Markus Forsberg, and Johan Roxendal. 2012. Korp - the corpus infrastructure of Språkbanken. In *Proceedings of LREC 2012*, pages 474–478, Istanbul, Turkey.

Johan Carlberger and Viggo Kann. 1999. Implementing an efficient part-of-speech tagger. *Software – Practice and Experience*, 29(9):815–832.

Hamish Cunningham, Yorick Wilks, and Robert J. Gaizauskas. 1996. GATE – a general architecture for text engineering. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Richard Domeij, Ola Knutsson, Johan Carlberger, and Viggo Kann. 2000. Granska – an efficient hybrid system for Swedish grammar checking. In *Proceedings of Nodalida '99*, pages 49–56, Trondheim, Norway.

Rickard Domeij, Joachim Hollman, and Viggo Kann. 1994. Detection of spelling errors in Swedish not using a word list en clair. *Journal of Quantitative Linguistics*, 1:195–201.

Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. 1992. The linguistic annotation system of the Stockholm-Umeå Corpus project. Technical report, Department of General Linguistics, University of Umeå (DGL-UUM-R-33), Umeå, Sweden.

Björn Gambäck and Fredrik Olsson. 2000. Experiences of language engineering algorithm reuse. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece.

Viggo Kann. 2010. KTHs morfologiska och lexikografiska verktyg och resurser (Morphological and lexicographical tools and resources from KTH). *LexicoNordica*, 17:99–117. QC 20120126.

Ola Knutsson, Johnny Bigert, and Viggo Kann. 2003. A robust shallow parser for Swedish. In *Proceedings of Nodalida 2003*, Reykjavik, Iceland.

Ola Knutsson, Johan Carlberger, and Viggo Kann. 2001. An object-oriented rule language for high-level text processing. In *NoDaLiDa'01 - 13th Nordic Conference on Computational Linguistics*, Uppsala, Sweden.

Jonas Sjöbergh and Viggo Kann. 2004. Finding the correct interpretation of Swedish compounds a statistical approach. In *Proceedings of LREC-2004*, pages 899–902, Lisbon, Portugal.

Jonas Sjöbergh and Ola Knutsson. 2005. Faking errors to avoid making errors: Very weakly supervised learning for error detection in writing. In *Proceedings of RANLP 2005*, pages 506–512, Borovets, Bulgaria.

# Pipeline for a data-driven network of linguistic terms

**Søren Wichmann**
Laboratory of Quantitative Linguistics
Kazan Federal University
wichmannsoeren@gmail.com

## Abstract

The present work is aimed at (1) developing a search machine adapted to the large DReaM corpus of linguistic descriptive literature and (2) getting insights into how a data-driven ontology of linguistic terminology might be built. Starting from close to 20,000 text documents from the literature of language descriptions, from documents either born digitally or scanned and OCR'd, we extract keywords and pass them through a pruning pipeline where mainly keywords that can be considered as belonging to linguistic terminology survive. Subsequently we quantify relations among those terms using Normalized Pointwise Mutual Information (NPMI) and use the resulting measures, in conjunction with the Google Page Rank (GPR), to build networks of linguistic terms.

## 1 Introduction

Linguistics is a discipline rich in terminology. Terminology specific to this domain is needed everywhere from the fine description of individual speech sounds over the categorization of different syntactic constructions to features of language use, and the abundance of terminology stemming from the empirical nature of inquiry itself is compounded by the excess of theoretical approaches, each of which tends to develop its own terminology. Thus, there is no dearth of handbooks of linguistic terms, but they only provide selective glimpses of the vocabulary coming into play when linguists write about languages. Here we take a data-driven (corpus-based) approach to the study of linguistic terminology using a set of 19,761 texts in English that belong to the DReaM corpus of linguistic literature (Virk et al., 2020). These texts consists of full grammars, partial descriptions of certain features, comparative studies, etc. That is, works that describe one or more features of the world's languages. According to the most recent count it spans 4,527 languages (Hammarström et al., 2021). It is important to emphasize that the corpus generally does not include purely theoretical literature. Thus, we are unlikely to come across some term that a theoretician has proposed if its actual usage in descriptions is rare.

This paper has two foci, where the first (1) is the pipeline immediately preceding the harvest of linguistic terms and the second (2) is the analysis of relationships among those terms. As for the first focus (1), we exclude a discussion of all the work that has gone into assembling the corpus, preparing metadata, and running documents through OCR. Instead, we focus on the pipeline for extracting linguistically relevant terms. This pipeline will be presented only summarily, but all steps, both trivial and less trivial ones, will be listed. The second focus (2) is on the relationships among terms. Mapping the relationships between these terms serves two purposes. First, (2a), the online DReaM corpus[1] currently allows for string searches in the available texts. We would like to enhance this functionality with an option to retrieve search results not only for a specific term but also related terms. For instance, in the procedure to be explained below, we find empirically that the term *direct object* is closely related to *indirect object*, and *relative clause* is closely related to *head noun*. A user should be given the option of choosing to include such related terms in a search. Secondly (2b), we want to analyze the network or networks constituted by related terms. A central question here is whether the network(s) can somehow lay the ground for an ontology of linguistic terms.

[1] https://spraakbanken.gu.se/korp/?mode=dream?lang=en

## 2 Related work

This work pertains to the fields of terminology extraction and automated domain ontology construction. Although the literature in these areas is rich (Medelyan et al., 2013; Qiu et al., 2018; Heylen and Hertog, 2015), it is not the case that an appropriate off-the-shelf tool can be found and applied to the case at hand. Most approaches are directed at cases which are more privileged in terms of the nature of the corpora analyzed. A large proportion of the texts of our sample are replete with OCR errors making the filtering of noise a real issue which is not usually present. Some approaches take recourse to generic resources such as WordNet for establishing concept relations or plugging relations into a wider framework (Navigli and Velardi, 2004; Alrehamy and Walker, 2018). Linguistic terminology, however, is of such a specialized nature that such resources cannot easily be drawn upon. Related to this problem, the common strategy of identifying hypernym-hyponym or is-a relations from texts (Velardi et al., 2004; Alfarone and Davis, 2015) is complicated by the abstract nature of linguistic terminology and the fact that many such relations depends on a particular theoretical framework. For instance, a *subject* can be a kind of topic, argument, position, noun etc. depending on the language, point of view, and theory of grammar. Moreover, such terms are often defined through examples rather than discursively in different grammars. Our approach is minimalist, so we also do not produce a fully POS-tagged corpus as input to term extraction, unlike some other approaches (Bourigault and Jaquemin, 1999).

There seems to be just one published approach similar to ours (Kang et al., 2016). It is similarly a minimalist approach, only relying on the particular corpus of interest. It proceeds from the extraction of terminology to a procedure of relating terms through a vector-based similarity metric. Nevertheless, this approach and ours are only comparable at a general level.

## 3 Pipeline for term extraction

The following describes the pipeline in numbered steps. Most steps were carried out using R, while a few steps additionally involved Python scripts.

**S1**. An initial database of text files OCR'ed from linguistic descriptive materials was used. These have been collected and processed by Harald Hammarström over several years (Virk et al., 2020). He also supplied a bibliography file in BibTex style with metadata (henceforth source.bib), which was parsed. The current version of this file is publicly available as part of Glottolog (Hammarström et al., 2020).

**S2**. When several files were associated with the same bibliographical entry, the besttxt field of source.bib was visited in order to select the best file.

**S3**. Files tagged in the bibliography as not primarily being grammatical descriptions, but rather lexicographic, ethnographic, etc. works, were removed.

**S4**. Works having English as the metalanguage (i.e., works written in English, although typically describing some other language) were singled out. Documents using a metalanguage other than English were removed.

**S5**. All lines having characteristics of something other than running text (tables, lists, short headings, bibliographical entries, etc.) were removed. A machine learning system for recognizing bibliographical entries is under development, but was not actually applied. Remaining lines were concatenated in a single line and subsequently split into sequences delimited by a full stop—in most cases representing sentences, but best described neutrally as 'chunks'. They were then put in a single file, collected.txt.

**S6**. Another file was created with two columns: one having numbers representing the sentence number in collected.txt and another having the file names. Thus, numbers indexing terms remain cross-referenced with the document where they occur.

**S7**. Since in linguistics, as in so many other domains, terminology is generally represented by noun phrases rather than just nouns (Nakagawa and Mori, 1998), an NLTK-based shallow parser (Babluki, 2013)[2] was used to identify noun phrases representing the topics (terms) of each sentence.

**S8**. The list of all terms and their indices was converted to a list of unique lower-cased terms, each with a list of indices. Most recently, this list had 34,437,644 items. Note that at this stage any term is included, not just linguistic terms. (Henceforth we will simply indicate new numbers of items in square brackets and preceded by an arrow as we go through the steps that it took to reduce the list).

**S9**. Only terms occurring 50 times or more were

[2] Available at https://gist.github.com/shlomibabluki/5539628

retained. [→ 142,729 items].

**S10-11**. Files were prepared allowing to determine the number of different documents in which a terms occurred. After manual inspection it was decided that a term should occur in at least 6 documents in order to minimize noise and maximize the inclusion of valid linguistic terms. [→ 133,927 items].

In the following three steps a rudimentary form of Named Entity Recognition (NER) is applied. The goal is to remove such entities not belonging to linguistic terminology.

**S12**. The presence of author names in the list of terms was reduced by matching more than 30k names found in source.bib with the list of terms. [→ 129,791 items].

**S13**. The presence of language names in the list of terms was reduced by matching more than 30k language names from an earlier version of Ethnologue (Eberhard et al., 2020) with the list of terms. [→ 121,699 items].

**S14**. The presence of publishers in the list of terms was reduced by matching more than 7k publisher names from source.bib with the list of terms. [→ 121,371 items].

**S15**. Manual inspection showed noisy terms to often have one of the following symbols in initial position: ‘,  /, ¡, =, ¿, @, , —, ,  , , $. Such terms were found and deleted. [→ 117,648].

**S16**. Since the number of terms was still very large, at this point we passed from just eliminating negatives (non-linguistic terms) to first identifying positives (linguistic terms). This was done by using a glossary of linguistic terminology (7819 terms, including spelling variants) from the Summer Institute of Linguistics (SIL).[3] 3684 out of the 7819 SIL terms were found to recur among the 117,648 surviving terms in a non-case sensitive matching. We reasoned that a bona-fide linguistic term should bear some distributional similarity to at least one member of the core set of 3684 verified linguistic terms. The amount of similarity could be used as a cut-off for excluding terms not likely to be linguistic in nature. Thus, we measured the Normalized Pointwise Mutual Information (NPMI) (Bouma, 2009) between each of the 117,648 extracted terms and each of the 3684 verified linguistic terms among them, isolating the highest value and using that as a criterion for ‘linguisticality’ of the term. Some manual inspection showed that a maximal NPMI value of 0.5 would allow for a good balance between the inclusion of true positives and computational feasibility. By settling on this cut-off we excluded 98,474 terms, leaving 19,174. The vast majority of the included terms are relevant for the field of linguistics, and a $19,174 * 19,173/2 = 183,811,551$ size object entering into the computation of all pairwise NPMIs (see next section) can be handled efficiently in R.

The list of 19,174 terms along with indices linking them to sentence-like chunks in the collective file containing our database of linguistic literature (further linked to bibliographical references and other metadata) constitutes the basic data for this study. Several steps in the pipeline could be improved. For instance, more work could be done (and is being done) on the identification of bibliographical references in the text, and improvements to and extensions of the NER steps are eminently possible. Moreover, steps taken preceding the pipeline on OCR-error correction and other improvements of the input will increase the performance as well. Finally, it would be helpful if some form of performance evaluation could be developed (Granada et al., 2018). Still, taking into account the likely presence of a few thousand false positives, we have arrived at a list of linguistic terms about twice as large as the handmade SIL list and, most importantly, the list is one that reflects actual usage.

## 4 Related terms

Given that the list of 19,174 terms is associated with indices representing their occurrence in texts we could compute NPMI values (Bouma, 2009) for all pairs (using our own implementation of the NPMI). Pairs receiving the value -1, meaning that they do not co-occur, were excluded from further consideration. We also computed the Google Page Rank (GPR) for each of the items using the R package igraph (Csardi and Nepusz, 2006). The textual units used for computing NPMI and GPR were the ‘chunks’ (mostly equal to sentences) mentioned earlier.

Analyzing and plotting networks based on these data are useful aids in coming to decisions both about the design of a search functionality involving related terms and the prospects of basing an ontology of linguistic terms on such networks. Figures 1-2 show two clusters of related terms, selected from 3537 clusters. Clustering is based on a two-

[3] Available at https://feglossary.sil.org/english-linguistic-terms (accessed 2019-09-02).

column table where each of the 19,174 terms sits next to the term to which it has the highest NPMI value, here called 'best friend'. The 3537 clusters were extracted using igraph[4]. They range from having 2 to 200 elements, with median size 3 and mean size 5.42. log(size) and log(rank-of-size) is roughly a power-law distributed function (fit: $R^2 = .964$, exponent: -.668). Figures 1 and 2, respectively, are rather typical of a simple and a more complex cluster. The size of a cluster is determined by the availability of neighbors. For instance, the best friend of *voicing* is *degemination*, but there is no term that has *voicing* as its best friend. And all the clusters contain exactly one knot, representing the situation where two terms are each other's best friends. In both figures an arrow indicates relatedness in terms of NPMI and the direction of the error is from the term with the higher GPR to the one with the lower GPR. These directions currently have no real functionality but are included for exploratory purposes.

The clusters tend to be tightly knit around particular areas of linguistic terminology, as in the terms in Figure 1 that refer to processes that consonants may undergo (typically in intervocalic position) and the terms in Figure 2 that refer to elements of the organization of narratives.

We believe that the kind of clustering approach illustrated in Figures 1 and 2 is a useful way of supplying a search machine with suggestions for search terms that are related to the target term. Another possible approach would be to pick the terms that are highest-ranked in terms of their NPMI value, but they would tend to occur in the text returned for the target term by the search machine anyway and would not take the user in new, yet related directions in the same way as the present approach. The choice of how many terms should be returned is a matter of design. Currently even all elements of the largest cluster (200 terms) can be accommodated in a drop-down menu, so no restrictions may be necessary. The order of such a list could be determined by closeness in terms of the number of connecting edges, ties being resolved by GPR values, for instance.

As for the prospects for developing an ontology of linguistic terminology we believe that the present approach could also be productive. The clusters identified already offer themselves as basic components. One challenge is to connect these clusters. It seems that this could be done by finding an 'NPMI friend' of an appropriate member of the cluster in another cluster, and then linking clusters through such single edges.

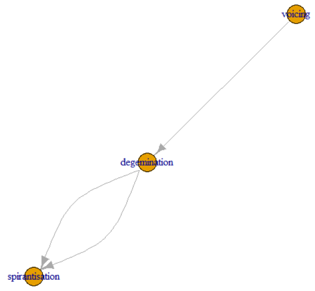


Figure 1: A simple cluster of related terms.

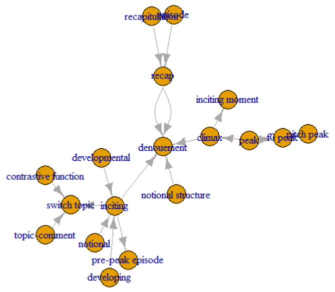


Figure 2: A more complex cluster of related terms.

## 5 Conclusion

In this paper we have demonstrated a pipeline for extracting terms from a thematically coherent text corpus, in this case a corpus of descriptive linguistic literature (to refer back to the outline in the Introduction this was Focus 1). We then went on to show that a simple clustering method, relying on single 'best friends' in terms of Normalized Pointwise Mutual Information (NPMI), is a useful basic step for designing a search machine suggesting search terms related to the target term (Focus 2a) and also has potential for helping in the construction of an ontology (Focus 2b).

[4] 'graph from edgelist' and 'decompose' functions

We place importance on the fact that the pipeline for the extraction of domain-specific terms was fully automated, apart from some shortcuts where we used list of terms from external sources to prune the list.

Future work not already mentioned above, will go into developing a more systematic evaluation procedure, applying a similar pipeline to texts in languages other than English, and connecting the output in a ways such as to create both a multilingual search machine and a multilingual ontology.

## Acknowledgments

Our research was carried out under the auspices of the project "The Dictionary/Grammar Reading Machine: Computational Tools for Accessing the World's Linguistic Heritage" (NWO proj. no. 335-54-102) within the European JPI Cultural Heritage and Global Change programme (http://jpi-ch.eu/). It would not be possible without the DReaM corpus and associated metadata painstakingly compiled by Harald Hammarström.

## References

Daniele Alfarone and Jesse Davis. 2015. Unsupervised learning of an IS-A taxonomy from a limited domain-specific corpus. In *IJCAI'15: Proceedings of the 24th International Conference on Artificial Intelligence*, pages 1434—1441.

Hassan H. Alrehamy and Coral Walker. 2018. Semcluster: Unsupervised automatic keyphrase extraction using affinity propagation. In *Advances in Computational Intelligence Systems: Contributions Presented at the 17th UK Workshop on Computational Intelligence, September 6–8, 2017, Cardiff, UK*, pages 222–235, Cham. Springer.

Shlomi Babluki. 2013. An efficient way to extract the main topics from a sentence. https://thetokenizer.com/2013/05/09/efficient-way-to-extract-the-main-topics-of-a-sentence/. Technical report.

Gerlof Bouma. 2009. Normalized (point-wise) mutual information in collocation extraction. In *Proceedings of GSCL*, pages 31–40. Gesellschaft für Sprachtechnologie und Computerlinguistik.

Didier Bourigault and Christian Jaquemin. 1999. Term extraction + term clustering: An integrated platform for computer-aided terminology. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 15–22, Bergen. Association for Computational Linguistics.

Gabor Csardi and Tamas Nepusz. 2006. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig. 2020. *Ethnologue: Languages of the World.* SIL International, Dallas, TX.

Roger Granada, Renata Vieira, Cassia Trojahn, and Nathalie Aussenac-Gilles. 2018. Evaluating the complementarity of taxonomic relation extraction methods across different languages. https://arxiv.org/abs/1811.03245.

Harald Hammarström, Robert Forkel, Martin Haspelmath, and Sebastian Bank. 2020. *Glottolog 4.3*. Max Planck Institute for the Science of Human History, Jena. (Available online at http://glottolog.org).

Harald Hammarström, One-Soon Her, and Marc Allassonnière-Tang. 2021. Keyword spotting: A quick-and-dirty method for extracting typological features of language from grammatical descriptions. In *Swedish Language Technology Conference 2020 (SLTC 2020)*. NEJLT.

Kris Heylen and Dirk De Hertog. 2015. Automatic term extraction. In *Handbook of Terminology, Vol. 1*, pages 203–221, Amsterdam. John Benjamins Publishing Company.

Yong-Bin Kang, Pari Delir Haghigh, and Frada Burstein. 2016. TaxoFinder: A graph-based approach for taxonomy learning. *IEEE Transactions on Knowledge and Data Engineering*, 28:524–536.

Olena Medelyan, Ian H. Witten, Anna Divoli, and Jeen Broekstra. 2013. Automatic construction of lexicons, taxonomies, ontologies, and other knowledge structures. *WIREs Data Mining Knowl. Discov.*, 3:257–279.

Hiroshi Nakagawa and Tatsunori Mori. 1998. Nested collocation and compound noun for term recognition. In *Proceedings of the First Workshop on Computational Terminology*, pages 64—70, Montreal. Université de Montréal.

Roberto Navigli and Paola Velardi. 2004. Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics*, 30(2):151–179.

Jing Qiu, Lin Qi, Jianliang Wang, and Guanghua Zhang. 2018. A hybrid-based method for Chinese domain lightweight ontology construction. *International Journal of Machine Learning and Cybernetics*, 9:1519–1531.

Paola Velardi, Stefano Faralli, and Roberto Navigli. 2004. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3):665–707.

Shafqat Mumtaz Virk, Harald Hammarström, Markus Forsberg, and Søren Wichmann. 2020. The DReaM corpus: A multilingual annotated corpus of grammars for the world's languages. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 878–884.

# Cross-Topic Author Identification – a Case Study on Swedish Literature

**Niklas Zechner**
Språkbanken
Department of Swedish
University of Gothenburg
niklas.zechner@gu.se

## Abstract

Using material from the Swedish Literature Bank, we investigate whether common methods of author identification using word frequencies and part of speech frequencies are sensitive to differences in topic. The results show that this is the case, thereby casting doubt on much previous work in author identification. This sets the stage for a broader future study, comparing other methods and generalising the results.

## 1 Introduction

Author identification is a competitive field, with many studies reporting ever increasing accuracies. Often, the accuracy as reported by the experiment is seen as irrefutable proof that the method works. But there may be reason to be sceptical of the optimistic results. Previous work has shown that there are several things to take into account for text classification generally, before methods can be considered reliable and comparable. The size of the texts has a large impact on the accuracy, and naturally the number of candidate classes also matters (Zechner, 2017). Even minor details in how the test data is handled can lead to significant overestimation of the accuracy (Zechner, 2014).

When it comes to author identification specifically, one of the main pitfalls is neglecting to account for differences in topic, style, or genre (Mikros and Argiri, 2007). If we apply a classification method to texts by several different authors, but each author mainly writes on a particular topic, how do we know if the classification method is detecting authors or topics? If the method is sensitive to topic, the accuracy reported in testing may be far higher than what we would get from a real-life application, where the text to be identified is on a different topic. Ideally, we would like to test this using texts marked for both topic and author, but performing such a study would be difficult at best – not only would it be hard to find a large corpus marked for topic, it is also doubtful if any two texts can be said to be on exactly the same topic.

The question of what topic really means is of course a matter of both debate and opinion, but that discussion is not really relevant here. For our purposes, we can essentially define topic as everything that is not author – any traits of a text which do not correspond to traits of the author can be considered effects of the "topic", including genre, medium, level of formality, and so on.

Many have tried to get around the problem by basing their methods on features of the text which are assumed to be independent of topic. Perhaps the most famous example is by Mosteller and Wallace (1964), in their study on the Federalist Papers. They based their analysis on the frequencies of function words, that is, words whose meaning is mainly grammatical rather than semantic, arguing that those words should not be dependent on topic. But they did not put that assumption to the test, and few have done since. While it may seem sensible to think that simple grammatical words like ”the” or ”of” should be used with about the same frequency across all topics, it is arguably just as sensible to say that they should be used equally by all authors.

Since it is unfeasible to find texts on the same topic by different authors, we have to approach the problem differently. One thing we can find is texts by the same author that can be considered different in topic, at least in this broad sense. Using a corpus of such texts, we can compare how well a method performs in different situations – is the accuracy lower when the texts we try to match up are on different topics? We can also apply the same method to identifying a topic among texts by the same author, which gives us another indication of how sensitive the method is to topic.

In a previous study (Björklund and Zechner, 2017), we investigated this problem by examining a set of novels, using each separate novel as an approximation of topic. In this study, we begin to expand on that work and apply a similar approach to a larger corpus, this time in Swedish.

As an alternative to function words, some have tried using features based on grammatical analysis of the words. Could the grammatical patterns of an author be less topic-dependent than their use of function words? Different studies have given conflicting results, finding such methods to be worse (Menon and Choi, 2011), equally good (Luyckx and Daelemans, 2005), or better (Björklund and Zechner, 2017). We apply a method using parts of speech alongside the word-based method to see if there are differences in how they relate to topics.

### 1.1 The problem

In a typical author identification task, we want to find which of a set of candidate texts is written by the same author as a given target text. To test a method on this task, we need a number of text samples, at least two of which are by the same author. One of the two acts as the target text, and one is mixed in with texts by other authors to form the candidates. We now have a set of candidate texts with one "true" candidate, the one which is actually by the same author as the target text, and some number of "false" candidates, which are by other authors. If the method correctly identifies the true candidate, it is considered successful. By repeating the experiment, we can estimate the accuracy of the method, that is, the probability of successful identification.

Commonly, when we test a method like this, we only have access to an unstructured text or set of texts by each of a number of authors. This could be articles or letters, or internet data such as forum messages or blog texts. This causes a problem when evaluating the test results. If the methods can reliably identify text samples from the same source, is that because they are written by the same author, or is it because they are on a similar topic? There is a risk that the methods look very accurate in a test setting, but are actually much less so when we apply them to a real-life problem.

### 1.2 The approach

To address this issue, we use text samples from books, under the hypothesis that each book can be seen as a separate topic. (Note, again, that we are using "topic" effectively as shorthand for "any systematic difference that is not directly due to the author" – genre, context etc.) This allows us to try three variants of the identification task, as illustrated in Figure 1.

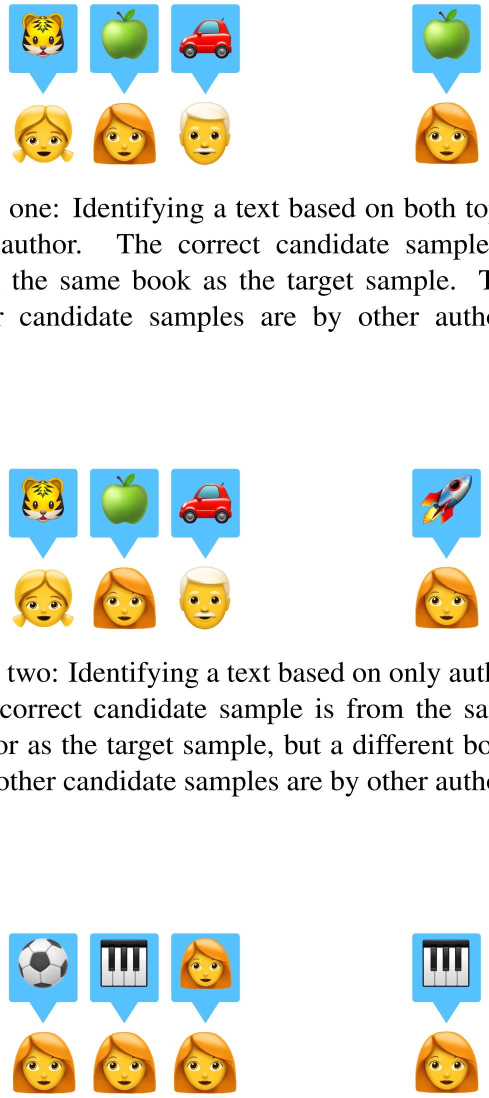
In the first case, the true candidate comes from the same book as the reference text, and the false candidates come from books by other authors. This corresponds to the commonly seen experiment, where we are effectively identifying a combination of author and topic. In the second case, the true candidate is from the same author as the reference text, but not from the same book, and the false candidates are again texts from other authors. This way, we are identifying author without the influence of topic. In the third case, the true candidate is again from the same book as the reference text, but the false candidates are now from other books by the same author. Now we are identifying only topic, without the influence from author. By comparing the results, we hope to see if the method is more sensitive to author or topic.

Using books also has the advantage that we get a large amount of text for each author and topic, which helps reach a reasonable accuracy with simple methods. We will not attempt to make the method as accurate as possible, but rather keep it simple and transparent. This is because the goal here is not promoting a method, but rather showing the effects of topic on existing methods.

## 2 Data

We use data taken from the Swedish Literature Bank (litteraturbanken.se), a collection of old novels, from which we include only the ones that have been manually digitised. We restrict the data to works in Swedish, by a single known author, and leave out works that contain duplicate text, such as multiple editions of the same book. This leaves 481 books by 140 authors.

Each book is cut up into pieces of 40 000 words, leaving out any trailing words. One reason for this is so that the texts are all the same lengths, making the results meaningful and reproducible. Previous work has found that the accuracy of classification varies greatly with the length of texts, so that if we were to include entire books of varying length, the experiments would have little predictive value (Zechner, 2017). Another reason is that we want to compare texts from the same book, so it is necessary to divide at least some of the books into parts. We get 825 pieces in total.



Case one: Identifying a text based on both topic and author. The correct candidate sample is from the same book as the target sample. The other candidate samples are by other authors.

Case two: Identifying a text based on only author. The correct candidate sample is from the same author as the target sample, but a different book. The other candidate samples are by other authors.

Case three: Identifying a text based on only topic. The correct candidate sample is from the same book as the target sample. The other candidate samples are from other books by the same author.

Figure 1: Illustration of the method.

## 3 Method

We use a feature set consisting of just ten (relative) word frequencies, specifically those words that are the most common in the data generally. “Words” here also include punctuation, and are counted independent of capitalisation. The words in this case are: comma, full stop, “och”, “i”, “att”, “det”, “en”, “som”, “han”, “jag”.

For each text (that is, for each piece of 40 000 words), we create a profile of its frequencies for these ten words. As a distance measure, we calculate the (absolute) difference in each feature value, and sum over all features; in vector terms, this is the Manhattan distance, without any normalisation. Using these profiles, it is easy to compare any pair of text and calculate the distance. That can then be applied to the identification problem as described above, by comparing the target text to each of the candidate texts, and choosing the one with the smallest distance measure.

### 3.1 Measuring accuracy

Now we can run the three tests we want to compare: identifying a book among a set of books by other authors, identifying a book among a set of books by the same author, and identifying an author among others by comparing with a different book by that author. By repeating the process, we can find an estimated accuracy for each case.

But it is possible to go a step further. We can think of each of the possible pairs of texts as being of one of three types: Same book, same author (but different book), and different author. From the 825 chunks analysed, we get in total 537 same-book pairs, 16 356 same-author pairs, and 323 007 different-author pairs. Since the method is simple and fast, we can easily go though all the possible pairs, and find the distribution of distance measures for each type of pair.

Knowing this distribution has great value in a practical application, because it allows us to calculate the probability that a pairing is of a particular type, and thus the probability that two texts are by the same author, or from the same book. But we can also use it to get a better estimate on the accuracy of the identification problem.

Suppose we want to identify the author of a given text out of 100 candidates, using one other text by that same author and 99 texts by other (not necessarily distinct) authors. This will mean one same-author comparison, and 99 different-author comparisons. Using the simplifying assumption that the similarity between a given text and a random text by the same author does not correlate with the average similarity between that given text and a random text by a different author, we do not need to investigate specific text samples one by one. Instead, we can think of it as a simpler statistical problem: For a given same-author pair, how likely is it that it will have a lower distance measure than each of a set of 99 different-author pairs?

To find out, we do not need to choose 99 random different-author pairs. Instead, we keep a sorted list of the different-author pairs. Choosing one same-author pair, we can use a simple binary search to see what fraction $f$ of the different-author pairs have a higher distance measure. Then, the probability of 99 of them having a higher distance measure is just $f^{99}$; this is the probability of this same-author pair being correctly identified. This is simple enough that we can repeat it for all the same-author pairs, and calculate the average accuracy, without having used any random subset.

### 3.2 Further variations

If we look closer at this corpus, we find that there is one author who is far more prolific than the others: August Strindberg. Our sample contains no less than 64 of his works, far more than any other author. Since the number of same-author pairs for an author increases approximately as the square of the number of works by that author, that means that he has a very large impact on the results – about three quarters of the same-author pairs are from Strindberg. This might skew the results, so we run the tests twice, with and without Strindberg.

This corpus also includes a grammatical analysis, so we can try using that as an alternative to word frequencies. In a similar manner, we now count the frequencies of the ten most common parts of speech (POS) (including, again, punctuation).

## 4 Results

The distributions of distance values for the three types of pairs are shown in Figure 2. We can see that the distance values for same-author pairs are lower than those for different-author pairs, as can be expected, but also that the values for same-book pairs are lower still. This immediately tells us that methods like this one would be highly topic-dependent. In this graph, the separation between the same-book curve and the same-author curve tells us how strongly the method reacts to topic, and the separation between the same-author curve and the different-author curve tells us how strongly it reacts to author. A small overlap between the same-author curve and the different-author curve would indicate a method which is good for author identification, whereas a small overlap between the same-book curve and the different-author curve would indicate a method which *seems* good if measured by traditional testing.

The same-book and same-author distributions for Strindberg have been separated out. We can see that they have much higher distance measures, meaning that his works would be much more difficult to identify. Evidently, Strindberg has a more diverse writing style than most; further speculation is beyond the scope of this study.

Figure 3 shows the results of applying the POS method. We see that the results are very similar. The different-author curve still overlaps considerably more with the same-author curve than with the same-book curve, in approximately the same proportions as in Figure 2.

Note that the axes are largely arbitrary; the POS method has higher distance values, because the most common parts of speech have higher frequencies than the most common words, and the y axis is adjusted accordingly due to normalisation. The difference in height and width of the curves between the figures is therefore irrelevant. Also note that while we can see slightly larger overlaps both ways in Figure 3, indicating a lower accuracy, that is also mostly beside the point, since we are not interested in maximising the accuracy.

As outlined in the previous section, we can use the distributions to calculate what would be the average accuracy of an identification test. We choose an identification task with 100 candidates, and try the three different cases: Identifying a book among books by other authors (identification based on both author and topic), identifying an author among others while using a different book as reference (only author), and identifying a book among other books by the same author (only topic). The resulting accuracies are shown in Table 1. We see that in the second case, when we remove the influence of topic, the result is considerably lower, which confirms that the method is not topic-independent. The third case is also on a similar level, suggesting that the sensitivity to topic is in some sense comparable to the sensitivity to author.

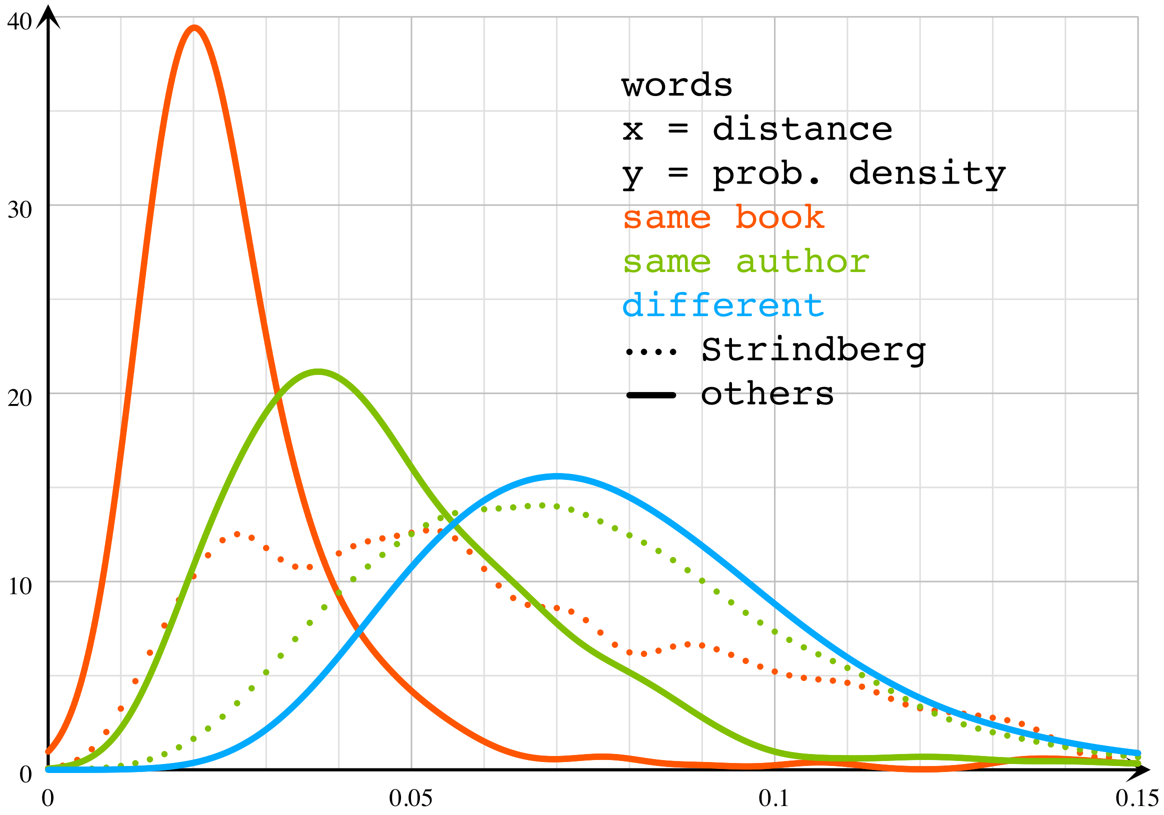


Figure 2: Distributions of distance measures for types of pairs. Distributions sum to one, and have been smoothed with a Gaussian blur, sd = 0.005. The different-author curve also includes Strindberg.

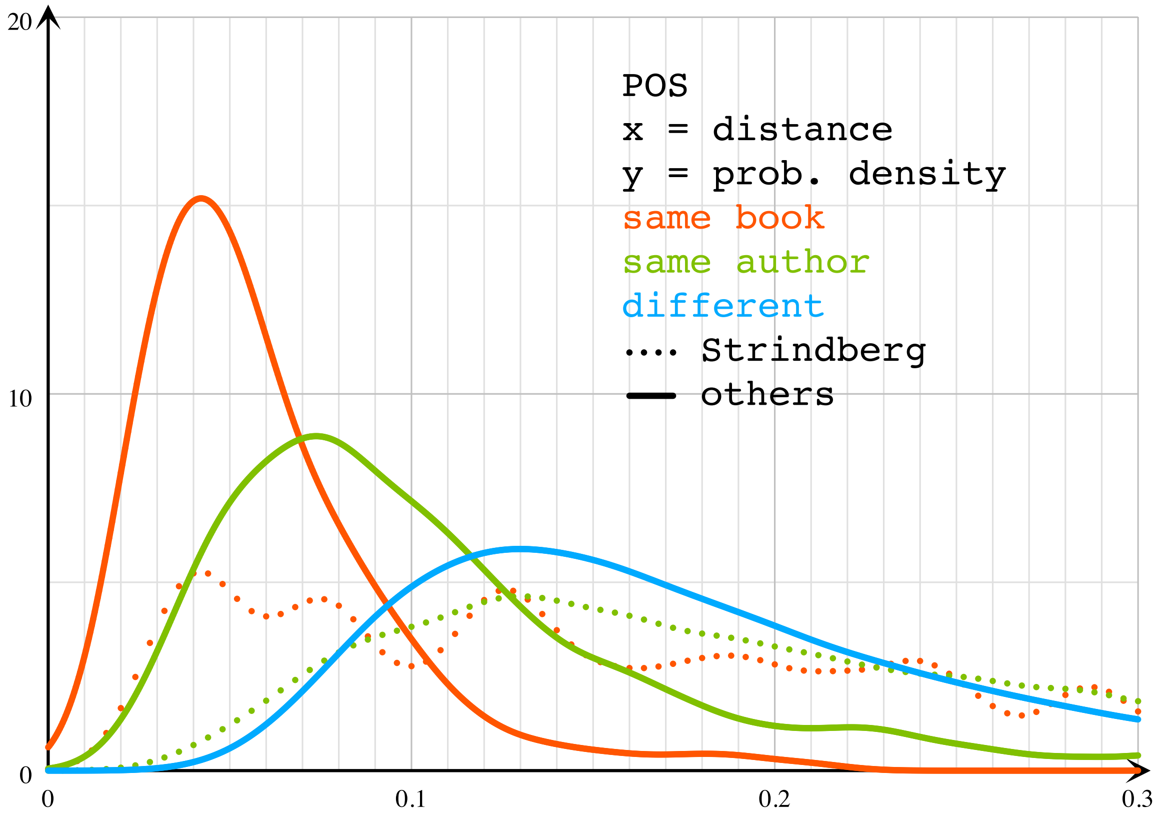


Figure 3: Distributions for POS features. Gaussian blur sd = 0.01.

| Comparison | Words | POS |
|---|---|---|
| **All authors** | | |
| same book vs. different | 52% | 43% |
| same author vs. different | 8% | 6% |
| same book vs. same author | 17% | 14% |
| **Without Strindberg** | | |
| same book vs. different | 67% | 53% |
| same author vs. different | 20% | 17% |
| same book vs. same author | 11% | 7% |

Table 1: Simulated accuracies for the different tests, for 100 candidates.

The distributions can also be used to calculate the probability that a pair is of a given type. For example, suppose we know that a text sample is either from book A, book B or book C. The three books are by different authors (neither of whom is Strindberg) and we have another sample of book A, but not of book B or C. We compare the unknown sample and the one from book A, and get a distance measure of 0.04 (using the word-based method). How likely is it that the unknown sample is from book A? Since there are three candidates, and we have no further information, the a priori probability is 1/3, or in other words, the a priori probability of a different-author pair is twice as high as that of a same-book pair. Looking at Figure 2, we see that at 0.04, the same-book curve is at 9, and the different-author curve at 6. The final probability for a same-book pair (and therefore, the probability that the unknown sample is from book A) is $1 * 9/(1 * 9 + 2 * 6) = 43\%$.

## 5 Discussion

We can see directly from the distribution curves that this method is not topic-independent. The accuracy calculations verify this, and indicate that the method may be at least as sensitive to topic as it is to author. This means that similar methods may not be reliable for author identification; even if experiments show promising results, the accuracy in a real-world application might be far lower.

We should keep in mind that this is not meant as a tool for topic identification; clearly there are far better methods for that. Whether this is an accurate representation of topic is also irrelevant, since we are interested in separating out any traits not related to the author. Furthermore, authors may well write several books on the same topic. But that would only mean that we have underestimated the problem. If we have only partially separated topic from author – as is almost certainly the case – the decrease in accuracy for a real application would be even greater. Future studies may be able to test this using data from more diverse sources.

It should be noted that the methods used here are not intended to be as accurate as possible. We could very likely improve the accuracy by using a larger set of features, or by using some form of normalisation on the feature values, or by using a more advanced classifier. It is also clear from tests not shown here that the accuracy depends heavily on the size of the samples; samples significantly smaller than these would drastically lower the accuracies, and larger samples would improve them. For the same reason, the overall difference in accuracy between the two methods also does not matter.

### 5.1 Comparison of methods

The difference between the analyses based on word features and POS features seems negligible, so these experiments did not reproduce the findings of our previous study on English novels (Björklund and Zechner, 2017). Looking at the results without Strindberg, the gap in accuracy between on the one hand the classic test (the first case in Table 1) and on the other hand the topic-controlled test (the second case) is 70% for words and 68% for POS – technically a better result for the POS method, but hardly compelling evidence of a difference.

Could a different set of features do better? The words used in the first methods were not chosen specifically to be function words, but it is clear that they are, just as most other common words. Clearly, using function words was not enough to ensure topic independence.

These words have no obvious relation to specific topics, and so there is no obvious way to choose less topic-dependent words. We also know that the amount of data used is a very important factor for accuracy, so unless the texts in question are extraordinarily large, choosing features other than the most common ones would lead to a significant drop in accuracy. Other common features used are word or character n-grams, that is, sequences of several words or characters. It seems quite clear that those would suffer from the same problems.

Different studies have also used many different classifier algorithms. While some would likely give higher accuracies than the simple one used here, we cannot reasonably expect that any other standard statistical measure or machine learning algorithm would be less topic-dependent when based on the same topic-dependent features. By using more opaque classifiers like those based on “deep learning”, or more opaque feature sets such as character n-grams, we also risk losing the ability to see what the classification choices are based on, which makes it harder to understand problems like that of topic dependence.

### 5.2 Future work

We hope to build on this small experiment towards a larger study of classification on this type of corpus. The large amount of data and clear metadata may be useful for other types of classification, including gender and year of writing. A more comprehensive study of different feature sets might also reveal which types of features are best for identifying authors, which are better for topic, and which are better for identifying something else entirely.

For a future method to be topic-independent, it would likely have to more explicitly address the issue, and separate topic features from author features. This is not in principle impossible; even in writing it is often possible to detect differences in dialect, age of the author, and other personal characteristics which will be stable across topics. Can we automatically detect which features are genuine author traits, or do we need to filter them manually? Can it be done for broad linguistic domains, or do we need to search for reliable traits in each application case separately? Can we expect to find enough such features to distinguish between large numbers of authors?

### 5.3 Conclusion

We have seen that the tests traditionally used to determine the accuracy of author identification methods fail to take into account the effects of topic, style, genre etc. This has led to an overestimation of how feasible author identification is in general. Our experiments give an approximation of a lower bound for that discrepancy, but it is not possible to say if the effects are actually even bigger. This calls into question under which conditions automatic author identification is at all a feasible problem, and shows the need for methods that are explicitly designed to avoid the pitfall of topic dependence.

## References

Johanna Björklund and Niklas Zechner. 2017. Syntactic methods for topic-independent authorship attribution. *Natural Language Engineering*, 23(5):789–806.

Kim Luyckx and Walter Daelemans. 2005. Shallow text analysis and machine learning for authorship attribution. *LOT Occasional Series*, 4:149–160.

Rohith Menon and Yejin Choi. 2011. Domain independent authorship attribution without domain adaptation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 309–315.

George K Mikros and Eleni K Argiri. 2007. Investigating topic influence in authorship attribution. In *PAN*.

Frederick Mosteller and David L Wallace. 1964. *Inference and disputed authorship: The Federalist*. Addison-Wesley.

Niklas Zechner. 2014. Effects of division of data in author identification. In *Proceedings of the fifth Swedish language technology conference*.

Niklas Zechner. 2017. *A novel approach to text classification*. Ph.D. thesis, Umeå universitet.