NTNU – Trondheim
Norwegian University of
Science and Technology

CLIRch, an extensible open source framework for query translation

evaluated for use on the Norwegian/Spanish language pair.

Morten Minde Neergaard

Master of Science in Informatics

Supervisor: Bjørn Gamback, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

CLIRch, an extensible open source framework for query translation — evaluated for use on the Norwegian/Spanish language pair.

Morten Minde Neergaard <m@8d.no>

May 10, 2012

NTNU – Trondheim
Norwegian University of
Science and Technology

Abstract

CLIR, Cross-Lingual Information Retrieval, is a field of research that can be highly useful in web search and for several other applications. Extensive research has been done on possible CLIR implementations, but as of yet there are no open source frameworks or applications readily available. The thesis focuses on building such a framework and evaluating it for use on the Norwegian/Spanish language pair.

The framework implemented uses query translation to submit queries to existing information retrieval (IR) implementations, and the framework itself holds no low-level IR algorithms. Experiments were performed on a small parallel corpus of Norwegian and Spanish texts, using the Xapian and PostgreSQL IR implementations. A comprehensive comparison of possible configurations was done, and certain measures were shown to be effective when searching for documents in either language.

The framework is implemented in a modular architecture, allowing the suggested additions and amendments to be implemented as add-on components. This is the main intent of the framework, and eases the process of building support for additional languages as well. For easing the adoption of the framework, additional components and data may be beneficial.

Some improvements are also possible for the tested language pair, through obtaining larger data sets or implementing certain language specific algorithms. Of particular interest is implementing effective decompounding of Norwegian compound words and phrase translation support. Suggestions are also made for how the system can be used to perform CLIR tasks in other languages.

Acknowledgements

First an formemost I have to thank all of you who have helped me get to where I am today. The people who have been there for me, believed in me, and given me inspiration. I thank you, for being who you are and for being a part of my life. My family, my friends, you mean the world and a half to me.

Perhaps more relevant to the thesis, I'd like give a big thank you to my adviser, Björn Gambäck. For all the help and advise, and moreover for being kind, helpful, intelligent and flexible. Also, a thank you goes out to all of the other helpful and interesting people working for the university — especially to Lars Bungum for all his help with the thesis. (Takk til Høiskolen, for å gjøre meg til studerende.)

Further I have to offer my thanks to both staff and students at the University of Buenos Aires. I learned a lot during my year abroad, both academically and personally. Special thanks go to the students of "apoyo escolar", for teaching me Spanish. (A los alumnos de apoyo escolar, gracias por enseñarme español.)

In this work, many external resources were indispensable. My love for open source software requires me to send a big thank you to the NLTK, FreeLing, the Oslo-Bergen Tagger project, PostgreSQL, Xapian and any other projects releasing their software as open source.

Some other universities in Norway have also been very helpful. Most notably, thanks go to UIB for their help in searching for a corpus, and to UIO for the NorNet data. Last, but not least: This work has been carried out in connection to, but not been financed by, the EC/FP7 project PRESEMT — grant agreement ICT-248307.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The Internet is playing a larger and larger role in our lives. The infrastructure it provides for exchanging information has made massive impact all around the world. The amount of information readily available is truly immense, and includes documents about just about any domain in a wide variety of written languages.

Modern Information Retrieval (IR) algorithms were born to ease the task of finding relevant information among a nontrivial amount of data. Solutions exist for indexing, searching and accessing databases and text collections of sizes from mere dozens of documents to the web itself. This is one of the key elements to how the Internet is used today.

1.1 The problem

The problem arises when you do not speak a world language, and/or there are other reasons why you cannot find the information you are looking for in a language you know. Recent surveys of languages on the Internet, such as [Q-Success, 2012] and [Miniwatts Marketing Group, 2010], indicate that over half the web sites in the world are written in English. At the same time roughly only one fourth of all Internet users are native speakers of English. This can imply that many users are forced to search for information in other languages than their own, creating a need for some kind of multi-lingual web search — making the Internet available to users with little or no English

skills. Furthermore, the statistics show that the percentage of web sites that are written in English is decreasing. Today, most of the Internet is available to a person that knows only English — this could, however, change.

In addition to the idea of making Internet content more readily available to more people, there exist many other cases where the language barrier needs to be crossed in Information Retrieval. International corporations may have databases with documents in multiple languages, libraries or other institutions may have books and publications in many of the written languages of the world. Enter Cross-Lingual Information Retrieval, CLIR.

1.2 The solution

CLIR refers to retrieval of documents that are in a language different from the one used to express the query. Much research has been done in making this possible, and much of it focuses on translating to and from English — this is a natural consequence of English being a very dominant language on the web.

For many languages, no CLIR research has been carried out. For others, research may exist but focuses on translating to or from English. The frameworks used in much of the available research also appear to be built ad hoc, and the software is generally never released publicly in any way. Few papers include implementation details beyond low-level algorithm specifics.

As there are few resources available for building a standardized CLIR application, especially using Open Source, this thesis will focus on the process of building such a framework. The aim of the framework is to be flexible and modular enough to support extensions for any written language and any specific algorithms that may be needed. All components built into such a framework should preferably be freely available as Open Source.

For evaluating the aptitude of the framework, the Spanish/Norwegian language pair has been selected. Some key characteristics of this language pair, considered before implementing:

- Rich resources are available for English, and much research has been done focusing on CLIR to and from English.

- As will be discussed, fewer natural language resources are readily available for the selected language pair.
- The author is familiar with multiple languages, among which English, Norwegian and Spanish are their strongest.

Implementing and testing functionality for this language pair helps ensure the framework is not biased towards English-centric functionality, and may give a more unique base of research. As will be shown, the limited resources available for these languages may make them more interesting candidates for translation, but can also prove an obstacle.

The goal of the thesis is not describing or implementing the state-of-the-art in CLIR. The main focus is that of creating a base framework that others can build on to run similar experiments, and investigating its suitability on the Norwegian/Spanish language pair.

1.3 Related work

One solution was found to be available, namely EXCLAIM — the EXtensible Cross-Linguistic Automatic Information Machine. EXCLAIM uses Wikipedia article data to do query translation, but the framework appears limited and undocumented. No results or details showing results obtained using EXCLAIM have been published, and from a quick inspection of the source code it appears to be very English-centric — only allowing translation of queries from English to a second language.

Some papers found from major conferences have referred to and/or used named frameworks. Especially UTACLIR (University of Tampere Cross-Language Information Retrieval) [Keskustalo et al., 2002], a dictionary based query translation framework, appears very relevant to this thesis. Again, however, no source code could be found and the framework appears not to have been used in any recent research.

Many articles have been found referring to CLIR systems between specific language combinations, showing results and techniques available for the language or language pair. Much of the research found has focused on English and other world languages, while some of it focused on how to implement CLIR for languages with limited digital resources available.

In later years research focus appears to have shifted towards more advanced and more specific multilingual software applications. The larger conferences discussing CLIR, e.g. the latest Cross-Language Evaluation Forum (CLEF) conferences [Forner et al., 2011], focus more on tasks such as image retrieval and plagiarism detection.

1.4 Thesis outline

Chapter 2 starts with a comprehensive review of CLIR and techniques applied within CLIR. An introduction to modern information retrieval algorithms is also given, as this is a key building stone in CLIR. This establishes a foundation for explaining the scope of the framework and this thesis, as well as the rationale for building the framework.

Then Chapter 3 presents the framework itself and its implementation details. The architecture is shown in a top-level overview before each component is introduced. The information retrieval algorithms used are also presented with all details found relevant to the evaluation.

In Chapter 4, the methods of evaluation are explained — the methods used as well as alternative evaluation forms. Details are given on the parameters used for all tests, before all findings are presented in Section 4.3. As the results are shown, focus is given to possible flaws and shortcomings in the framework or configuration.

Finally, Chapter 5 goes into what improvements can be done to the framework. Amendments and additions are discussed for the chosen language pair, and the inclusion of other languages is covered before Section 5.2 gives a closing statement.

Chapter 2

On Cross-Lingual Information Retrieval

The following list defines some possible assumptions for a system for CLIR based on query translation. Other base definitions are possible, but for the sake of clarity these assumptions will define the system limits of a CLIR system in the thesis.

- A query is formulated in one specific language, and this language is known by the CLIR system.
- The query is translated to a single, predetermined target language.
- The output query does not have to be a human readable, word-by-word translation.
- The IR engine is considered a separate system.
- The user understands the results returned by the IR system.

This chapter describes many techniques that have been found useful in CLIR applications, but does not go into detail on all aspects of any subject. References to more thorough articles are included. For a more comprehensive overview of both IR and CLIR, the book *Information Retrieval — Algorithms and heuristics* [Grossman & Frieder, 2004] could be a good starting point.

2.1 Crossing the language gap

The key challenge in any CLIR system is to cross the language gap, which is not as simple as just finding a dictionary and looking up the query. This section first details some of the processing that can, and often needs to, be performed on the input query. It then details the use of dictionaries in CLIR, before presenting some techniques that can be used to improve results.

2.1.1 Processing input

Before a word can be looked up in a dictionary, it needs to be split into words and these words must be normalized morphologically. For English, splitting a sentence into words can often be as simple as splitting it by whitespace or other simple metrics, often referred to as tokenization. As will be discussed, this is not the case for Norwegian and certainly not a possibility for some other languages — as an example, some Asian languages have no word boundaries in their written form.

With morphological normalization we mean algorithmically determining a base form or stem of any given word. Stemming algorithms such as the Porter stemmer [Porter, 1980] have been used extensively in IR, and function under the rationale as follows — exemplified in a simple, monolingual setting.

Assume a document D containing the words "radical coolness" and a query Q containing the words "radically cool". A search using this query would not find the document because the inflections differ! A stemmer reduces this problem to finding a root form that is identical for all inflected forms of the word. A stemmer might reduce both D and Q to the same form — "radic cool". The goal is to find a common stem for related words, not finding the canonical form.

Enter lemmatization. Lemmatization tries to find the canonical form of a word, its *lemma*. In many systems, e.g. when looking up a word in a dictionary, we are much better served using a lemma (e.g. "radical") than a stem ("radic"). Both stemming and lemmatization are language dependent tasks, but lemmatization often takes more factors into account — it may for example try to disambiguate or expand terms based on factors such as part of speech.

Without performing lemmatization, inflected forms might not incur hits in the dictionaries used. For some languages, in particular English, simple lemmatization approaches implemented in a manner similar to stemming may be enough to allow dictionary lookup. This is not true for many written languages — features such as variations in word stems and complex compounding call for language specific algorithms.

When lemmatizing languages with productive compounding, such as Norwegian, decompounding can be useful. Normally, roughly ten percent of the words in Norwegian text are compound words [Johannessen & Hauglin, 1996]. Thus, even though methods for decompounding Norwegian are quite complex, they can prove very useful. Methods for decompounding Norwegian exist and are highly reliable [Ranang, 2010].

Spanish does not have this productive compounding. It does, however, merge some pronouns into verbs so methods for separating or removing these compounded pronouns might be beneficial. For some details on the algorithms that have proven helpful for a series of languages, see an article written about the challenges in making CLIR systems for English, French, Arabic, German and Chinese — [Levow et al., 2005].

2.1.2 Dictionary usage

Although lemmatization can be a useful tool to find an entry in a dictionary lookup, it can have unwanted side effects — removing salient information from a word. A possible method for reducing information loss is backoff translation [Oard et al., 2000]. Backoff will, in the context of lemmatization, try to translate the unprocessed term before trying the lemmatized form. Backoff can also be applied to several parts of a CLIR system, including decompounding [Yang & Kirchhoff, 2006].

Once the terms have been processed, they can be translated to the target language. This is often done using a machine readable dictionary (MRD), in its simplest form a bilingual term list. Size and coverage can vary greatly between dictionaries, and the size of the dictionary is often a good metric for measuring its usefulness. Research indicates that for English a dictionary of 20,000 words or more is preferable [Demner-Fushman & Oard, 2003].

In the absence of an MRD for a given language pair, there exist ways of creating one. Such methods include using one or more pivot dictionaries, i.e.

translating via different languages [Gollins & Sanderson, 2001]. Other approaches include bilingual term list generation using parallel[1] or comparable text collections, or machine translation based on rules or statistics.

One method for generating a bilingual term list using parallel text is detailed in [Cancedda et al., 2003]. Using a corpus containing direct translations of each document, it finds words that tend to co-occur. If a set of English sentences contains the word "city" and the direct Spanish translations contain the word "ciudad", these terms are assumed to be translations of one another.

Wordnets can also be useful in a CLIR application. A wordnet maps word and meaning relations in a large network, using relation types such as synonymy and hyponymy. Beyond simple uses such as synonym lookup, there exist wordnets that map such relations across languages. Research and plans have been made for building a global grid of wordnets [Fellbaum & Vossen, 2007]. Such a wordnet might prove a useful data source for a CLIR application.

2.1.3 Choosing relevant translations

When translating any query term, potentially expanding using a synonym dictionary, multiple possible translations are yielded. Simply searching using all terms often results in a high number of false positives, as higher weight is given to terms with more translations [Levow & Oard, 2002]. Therefore, many techniques have been developed to determine which word or words serve as the best translation.

One such pruning technique was presented in [Federico & Bertoldi, 2002]. It uses statistical information about the target language to calculate the probability of each translation at a query level. The N most highly ranked query translations were eventually used for the search. Experiments showed that using only one translation often performed best.

For certain language pairs, reverse dictionary pruning has been shown to be efficient [Aljlayl et al., 2002]. When using this technique, first all terms are translated to the target language while keeping all candidate translations. Each term is then looked up in a dictionary translating back to the original language — only terms that translate back to the original term are kept.

Even after such techniques have been applied, the resulting terms may not

[1] Parallel texts are further described in Section 2.3.2

be the most relevant ones to the targeted documents. It is possible to amend this *after* presenting the user with search results, using relevance feedback. Relevance feedback means that the user selects documents that are good matches to their query, and statistics from these documents are used to expand or improve the query.

Also of possible interest is the pseudo-relevance feedback technique. It is based on the assumption that the first documents are likely to be relevant, and automatically uses them to improve the query. Other approaches also exist for improving IR results in the face of ambiguity — Section 2.3.3 covers methods that improve this by modifying IR algorithms or IR system usage.

2.2 Information Retrieval, IR

Modern IR can be split into a set of tasks to be performed on the corpus to be searched and on the queries made against it. For example, many of the basic input processing algorithms detailed under Section 2.1.1 are also relevant to IR systems. A key element to understanding the core workings of modern IR implementations is the vector space model, as introduced in [Salton et al., 1975]. This model represents documents as vectors, vectors where each dimension is a term found in the document collection.

A document collection containing D documents and a total of N distinct terms can thus be represented as D vectors, each a N-dimentional coordinate. The model allows calculations of similarity between two documents, or between a document and a query — these calculations are highly computationally efficient, but do not take word order into account.

Another limitation in the plain vector space model is that it weights all term overlap equally. As an improvement on this, the inverse document frequency (IDF) was introduced. In brief, IDF introduces a penalty for words that occur in all documents. Since its introduction in [Spärck Jones, 1972], the method has remained important in modern IR system implementations.

The vector space mode is a key building block to IR systems, but not the only component typically found in such a system. For more details on modern systems implementing these methods, see details of the Okapi BM25 and similar algorithms [Robertson & Zaragoza, 2009]. BM25 is a retrieval algorithm which has shown very good results at the annual Text REtrieval

Conferences (TRECs) from the mid nineties, and is one of the systems used for performing IR tasks in this thesis.

2.3 IR in a Cross-Lingual setting

2.3.1 Query and document expansion and translation

When building a CLIR system, a key design decision is whether one should use document expansion or only query expansion. One possible extreme is the simplest, using only query expansion and translation. With this approach, the system produces complete, translated phrases. These are matched against the target documents using a monolingual IR algorithm.

The opposite would be to focus on document expansion. It is possible to pre-process and translate the entire document collection at index time. Using this approach, the query is never translated but simply matched against the index that was built using translated terms. In systems with a large amount of documents, this can quickly consume nontrivial amounts of resources. It was originally proposed for spoken document retrieval [Singhal & Pereira, 1999], but has also been implemented for CLIR.

When expanding a query or document, one may choose to expand terms before or after translation, e.g. through a synonym dictionary. Effects are comparable when expanding terms post-translation (in a system using query translation) and when expanding terms pre-translation (in a system using document translation) [Levow et al., 2005].

Using document expansion requires control of the internals of the retrieval engine. This technique can not be used if one wants to connect the CLIR system to an external IR system.

2.3.2 Parallel corpora

A parallel corpus contains a collection of documents as shown in Figure 2.1. Each collection ($C_1..C_n$) has documents in one language ($L_1..L_n$). Each document collection holds the same (or very strongly related) documents ($D_1..D_m$), ($D'^n_1..D'^n_m$) in each language.

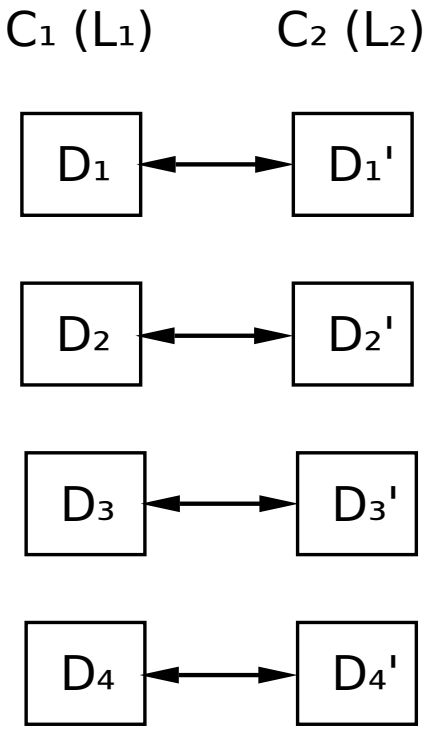

Figure 2.1: A parallel corpus with two languages and four parallel documents

If a corpus is built in a similar manner, but the texts are not direct translations of one another, the corpus is referred to as a comparable corpus. A set of articles from encyclopedias written in different languages is a good example of a comparable corpus — using Figure 2.1, D_1 and D_1' might be articles about Norway written in L_1 and L_2 respectively.

As mentioned in Section 2.1.2, such corpora can be used to generate bilingual term lists. They may also be useful in an evaluational setting, or as training data for a great number of statistical models. Several methods using parallel or comparable corpora as training data are detailed in this thesis.

2.3.3 Relevant IR extensions

When passing a translated list of terms to an IR engine without further processing, the system is using what is sometimes referred to as unbalanced

queries. An unbalanced query can contain a different amount of translations for each term, thus giving more weight to terms that yield more translations.

Balanced queries, as described in [Levow & Oard, 2002], try to distribute the weight given to each term in order to reflect the number of occurrences in the original query. Balanced queries can be implemented by passing some terms multiple times to the IR system. A similar approach is using structured queries [Pirkola, 1998], which make adjustments to wider parts of the weighting scheme directly in the IR engine.

There also exist an array of approaches to CLIR that use the vector space model to perform some type of latent semantic analysis. These approaches use parallel corpora and their vector space representation to automatically find related terms across languages. Such analyses can be used to implement a query expansion and translation system [Sahlgren & Karlgren, 2002], or to build an index with which one query could find documents in multiple languages [Dumais et al., 1997].

Chapter 3

Implementation

This chapter describes the framework that was implemented: "CLIRch" ([klɜːtʃ], near-homonym to search). The framework does query translation, transforming the query from one language to another. The chapter starts with an overview of the methods and architecture used, before going into implementation details.

3.1 Overview

As mentioned, CLIRch aims to implement query translation — translating individual queries from one language to another. Translations are performed by a chain of functions (Python *callables*) executed in a pre-defined order. These chains of functions will be referred to as *pipelines*.

Typically, at least one such pipeline exists for each language pair selected. Modules can be selected and configured to suit the languages chosen. A natural sequence of modules has been exemplified in Figure 3.1, showing a query being input in one language, processed, and output in the document language. A CLIRch configuration file can specify the contents of the pipelines per language pair, some of the modules in the figure are not obligatory.

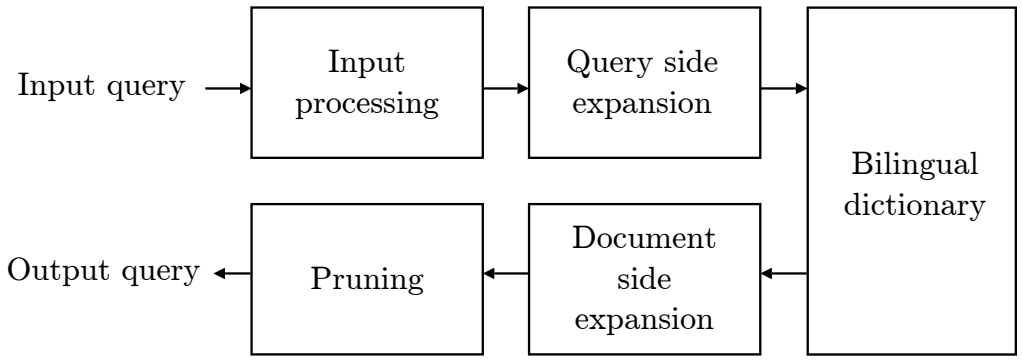

Figure 3.1: The minimal components used in a CLIRch pipeline

3.2 Components

CLIRch is separated into five modules. Each module holds components needed for building translation pipelines, and the user can combine these components as needed using configuration files or interactive manipulation. The following subsections detail each module. Only the implementation details that are of relevance to the thesis have been included in the descriptions.

3.2.1 The core module

The core module contains the base classes used for all user interaction, and code for loading custom configurations at runtime. It serves as glue for the remaining modules and exposes all basic functionality to the user.

3.2.2 The language module

Contains all code for registering language specific configurations. All languages are represented as instances of the `Language` class, which must specify the following five attributes:

- **Name** — Human readable name of the language, e.g. *English*.
- **Code** — Two-letter language code defined by ISO 639-1, e.g. *en*.

- **Preprocessing pipeline** — Sequence of callable objects to run before the query is translated, e.g. lemmatizer and stopword filter.
- **Translation pipeline** — Callable dictionary or dictionaries.
- **Postprocessing pipeline** — Pruning methods or morphological processing to be applied before outputting the terms.

3.2.3 The lookup module

The dictionaries — interlingual and intralingual — and utilities surrounding these. Any dictionary implementation must provide an interface for choosing languages based on ISO 639-1 language codes, as well as some other common options. The utilities in this module include functions for converting dictionaries to serialized forms and parts of the code needed for backoff translation.

3.2.4 The pruning module

This module contains the pruning methods available for selection in the postprocessing part of the pipeline. There are two notable pruning methods, the reverse dictionary pruner and the N-reduction pruner.

The reverse dictionary pruner

This pruning method prunes all terms that do not exist in a dictionary which translates back to the original language. This method has been successfully used for the English-Arabic language pair [Aljlayl et al., 2002].

The N-reduction pruner

This pruning method gives a ranking to all terms and discards all but the N most highly ranked. The ranking is done as follows:

- All terms are grouped by their origins.

- The groups are sorted by count of each term in the group.
- The sort is stable. This ensures that the order from the target language portion of the dictionary is kept.
- For each group, the N first terms are returned.

To exemplify, we will detail the processing of the Spanish phrase *error inolvidable* (unforgettable mistake) after translation to Norwegian. It uses the *wnes-red3* configuration for translating, as detailed in Section 4.2.

- All terms are grouped into two lists, one for terms originating from *error* and one for *inolvidable*.
- For *inolvidable*, there is only one translation in the dictionary — *uforglemmelig* (unforgettable).
- For *error*, there is a large amount of synonyms and translations. After passing through the complete translation pipeline, the four most frequent terms are *feil* (error, 17 instances), *feiltagelse* (mistake, 15), *feilaktighet* (incorrectness, 10) and *avvik* (deviation, 10).
- *avvik* is sorted after *feilaktighet* because it was encountered after *feilaktighet* in the dictionary.
- If the pruner is set to keep three terms, *avvik* and all terms with less instances are pruned.

3.2.5 The utils module

The utils module contains several components that are needed to work with natural language. This includes handling of different character encodings, stopword filtering and lemmatization/tokenization. Some of these components use functionality and data directly from the Natural Language ToolKit for Python, NLTK [Bird et al., 2009].

Tokenization/Lemmatization

All tests run against the system in this report use one of the following three components for processing the input queries:

- The simple tokenizer — uses a regular expression to find all sequences of word characters. Respects double quotes, treating their contents as phrases or named entities. Imported from the NLTK.
- The Norwegian Multitagger, a core component of the Oslo-Bergen Tagger [Johannessen et al., 2011]. Used to find all possible lemmata of an inflected word. Can detect some named entities.
- The Spanish tagger from FreeLing [Padró et al., 2010]. Used to find the most likely lemma of an inflicted word. Can detect some named entities.

Stopword elimination

The stopword eliminator indiscriminately removes all terms found in the relevant stopword list. The stopword lists are used directly from the NLTK. They originate from the Snowball stemmer, specifically the implementation used by PostgreSQL [PostgreSQL, 2008].

3.3 IR implementations used

3.3.1 PostgreSQL

The PostgreSQL DBMS comes bundled with full text search library, an open source text indexing and search implementation using well documented matching and ranking schemes. A number of customization options exist, but the tests run here have used only the default options — Snowball stemming and Snowball stopword lists.

The default ranking algorithm uses a simple term frequency similarity measure, normalized by dividing the rank by 1 + (the logarithm of the number of unique words in the document).

A second ranking algorithm is also available in PostgreSQL, namely cover density ranking [Clarke et al., 2000]. This is an approach to search ranking that includes physical proximity between the matched terms in the weighting of documents. It was thought to give the experiments a wider base of

comparison, and implementing it was not time consuming given the existing PostgreSQL implementation.

3.3.2 Xapian

Xapian is an open source search engine library written in C++. It has bindings for many modern programming languages, is under active development and uses state of the art retrieval algorithms such as Okapi BM25. For searching the corpus, some code was written based on the examples distributed with the Python bindings[1]. To structure named entities as phrases when appropriate, some changes were made to the example code.

For the BM25 weighting scheme, a custom extension was made. This extension uses information from the CLIRch pipeline to assign appropriate weights to terms. It currently only uses the weights assigned by the term frequency pruner described in Section 3.2.4. The result is weight assignments consistent with what is referred to as balanced queries (see Section 2.3.3).

The translations reached when translating *error inolvidable* in Section 3.2.4 were *uforglemmelig* (1 instance), *feil* (17 instances), *feiltagelse* (15) and *feilaktighet* (10). The weight assignment would separate these by source language term and distribute the weight among the target language terms. *uforglemmelig* would recieve the weight 100%, as it is the only translation from *inolvidable*. There are 42 result instances for *error*, resulting in weights *feil:*$\frac{17}{42}$, *feiltagelse:*$\frac{5}{14}$ and *feilaktighet:*$\frac{5}{21}$.

[1] `http://xapian.org/docs/bindings/python/examples/`

Chapter 4

Evaluation and results

Several methods exist for evaluating CLIR. A much used evaluation method is direct comparison with monolingual information retrieval. In this evaluation method, each query has a gold standard equivalent translation for each language. To evaluate, automatic translations are made from the gold standard queries. The search results obtained using the gold standard translations are compared to those obtained using the automatic ones using standard methods for IR evaluation.

The most relevant standard IR evaluation methods include precision, recall and F-measure. Precision is the percentage of documents returned that are considered relevant. Recall is the percentage of all relevant documents that is returned. F-measure combines precision and recall into one measure, e.g. using a harmonic average. All these metrics are often presented and compared using graphs.

The main problem encountered using this approach is that of finding a suitable corpus and queries with corresponding relevant texts. Finding a suitable Norwegian corpus with predefined queries proved difficult, and even for Spanish such corpora are not readily available. This means that anyone wanting to do such an evaluation might have to purchase or somehow create such data manually.

Further problems include the question of comparability between CLIR/IR results. Depending on the method used, the characteristics of a monolingual search result can be different from that of a CLIR search result. A naïve CLIR approach may yield poorer performance than monolingual IR, but due

to the nature of a system based on query expansion it may also yield results in excess of 100%, as shown in [Levow et al., 2005]. This can be caused by the synonym expansion that, implicitly or explicitly, occurs in the translation pipeline.

Parallel corpora can be helpful in CLIR evaluation. Given a query and a set of relevant texts in one language, a gold standard translation to a different language should relate to the equivalent subset of texts in its own language. Note, however, that even given a gold standard equivalent translation in both queries and documents, the inherent subtleties and ambiguities of natural language ensure that any query or document translation can introduce false positives and false negatives.

4.1 Corpus and evaluation method used

For the selected language pair, a very small and diverse corpus was obtained. It contains a set of documents, all originally written in Norwegian. Each has a Spanish counterpart, professionally translated. Documents are mainly fiction, and deal with extremely varied subjects.

Among the documents are complete novels of up to roughly 300,000 words, short stories shorter than 1,500 words, informational brochures, epilogues to Norwegian theatre plays, and more. A total of 31 documents are used, and their average length is roughly 50,000 words. Writing style, vocabulary, sentence length, etc. also varies greatly. Further details about the texts were not deemed particularly relevant, but may be made available upon request.

Because of the nature of the corpus, it was not possible to craft queries that were relevant to sizeable sets of documents. Therefore, a simpler evaluation method was used. To find potential weaknesses in the CLIR methods used, a large set of queries was written. Each query was written to be relevant to only one document. Using such queries, any precision, recall or F-measure measurements could be reduced to two numbers: The ranking at which the relevant number is returned, and the total number of documents returned.

Only the query/document combinations with the highest exemplary value were chosen for discussion — i.e. those which were successful in illustrating system weaknesses.

4.2 Experimental setup

This section details all the configuration parameters used in the tests that have been run. As it proceeds, it will give some explanations as to why exactly these modules and parameters have been chosen for running the tests.

4.2.1 Configuration parameters

Input processing

Data is always forced to a unicode representation, and stopword removal is always applied. Morphological processors are used as detailed in each configuration. None of these components require any configuration.

Dictionary

Two different dictionaries were used for the experiments. The one used predominantly is proprietary, obtained in XML format directly from the publishing company. The original dictionary files contained grammatical information, example translations and other metadata. It was, however, only used as a bilingual term list — mapping simple terms and phrases in a one-to-many relation. This dictionary held 22,165 Spanish terms with an average of 3.1 Norwegian translations, and 24,277 Norwegian terms with an average of 1.97 Spanish translations.

The other dictionary used was a freely available online dictionary. This dictionary held no phrases, examples or metadata and was somewhat smaller than the XML-based one. The online dictionary held 20,979 Spanish terms with an average of 1.62 translations and 21,817 Norwegian terms with an average of 1.57 Spanish translations.

Neither dictionary will be named here, for several reasons. Firstly, their names are not of any value to the discussions at hand, and may as such be omitted. Secondly, pointing out strengths or weaknesses of either dictionary in this context could be harmful to the dictionaries' owners. Thirdly, naming the dictionaries could indicate a bias towards the specific company

or product. Lastly, should anyone be interested in knowing the names of the dictionaries, or details on how they were processed, this will be made available upon request.

Pruning

All pruning techniques used are detailed in Section 4.2.1. For the reverse dictionary pruner, there is no configuration beyond dictionary selection. For all examples involving this pruning method, the same dictionary was used for translating terms back either way.

For the N-reduction pruner, a key decision was how many terms were to be kept. Rather than deciding fixed parameters for this setting, tests were run using one through nine terms as the cutoff. Only results from one to four will be displayed, as higher numbers failed to yield improved or interesting results.

Synonyms

For both Norwegian and Spanish, synonym expansion was implemented. A free Spanish wordnet was found distributed with FreeLing [Padró et al., 2010], and could easily be included — senses in this WordNet were extracted from EuroWordNet and distributed under GPL. For Norwegian, the NorNet wordnet [Fjeld & Nygaard, 2009] was obtained through direct contact with the UIO. It should be noted that the NorNet is a work in progress and focuses on noun relations.

4.2.2 Running the tests

The tests were all performed in a single run using the CLIRch framework. All configurations used were written to configuration files, and these were loaded and used sequentially. A small ad hoc script was made for running the translations, searching through the corpus and logging the results.

4.2.3 Tables of configurations

All test configurations have been detailed in Table 4.2, and Table 4.1 shows the search engines used. These tables summarize the components and configurations used, identifying them with short identifiers. Each identifier tries to summarize the details of that configuration in a compact way — e.g. `wnes` specifies that the Spanish WordNet from FreeLing was used. These identifiers will be used extensively in the following sections for the sake of brevity.

Table 4.1: IR engines and weighting schemes used

Engine	**Implementation**	**Ranking scheme**	**Extensions**
`Pg-trad`	PostgreSQL	Traditional	
`Pg-cd`	PostgreSQL	Cover Density	
`Xap`	Xapian	BM25	
`Xap+w`	Xapian	BM25	Balanced queries

Table 4.2: Configuration sets used

Configuration	**Lemmatizer**	**Dictionary**	**Pruning**	**Synonyms**
`red1`	FreeLing/OBT	Proprietary	1-Reduction	None
`red2`	FreeLing/OBT	Proprietary	2-Reduction	None
`red3`	FreeLing/OBT	Proprietary	3-Reduction	None
`red4`	FreeLing/OBT	Proprietary	4-Reduction	None
`nolemtz`	None	Proprietary	1-Reduction	None
`reverse`	FreeLing/OBT	Proprietary	Reverse	None
`webdict`	FreeLing/OBT	Web-based	1-Reduction	None
`wnes-red1`	FreeLing/OBT	Proprietary	1-Reduction	FreeLing
`wnes-red2`	FreeLing/OBT	Proprietary	2-Reduction	FreeLing
`wnes-red3`	FreeLing/OBT	Proprietary	3-Reduction	FreeLing
`wnes-red4`	FreeLing/OBT	Proprietary	4-Reduction	FreeLing
`wnno-red1`	FreeLing/OBT	Proprietary	1-Reduction	NorNet
`wnno-red2`	FreeLing/OBT	Proprietary	2-Reduction	NorNet
`wnno-red3`	FreeLing/OBT	Proprietary	3-Reduction	NorNet
`wnno-red4`	FreeLing/OBT	Proprietary	4-Reduction	NorNet
`wndb-red1`	FreeLing/OBT	Proprietary	1-Reduction	Both
`wndb-red2`	FreeLing/OBT	Proprietary	2-Reduction	Both
`wndb-red3`	FreeLing/OBT	Proprietary	3-Reduction	Both
`wndb-red4`	FreeLing/OBT	Proprietary	4-Reduction	Both
`wndb-reverse`	FreeLing/OBT	Proprietary	Reverse	Both
`wndb-webdict`	FreeLing/OBT	Web-based	1-Reduction	Both

4.3 Results

4.3.1 How to read the results

For the sake of readability, tables containing details for each query/document combination have been placed in Appendix A. Results for document D_1 through D_7 have been labeled Table A.1 through Table A.7, respectively. Summaries of all experiments are included in two tables, Table 4.3 and Table 4.4.

The retrieval result tables all use the same layout for showing results. One axis lists the IR engines used and the other lists the CLIRch configurations tested. For each of these engine/configuration combinations a translation and a retrieval run was performed, and the result summarized in the corresponding cell. Retrieval using the gold standard translations have been included as the top row of each table.

Each cell contains the simplified evaluation metric — the number of false positives ranked higher than the target document, and the total amount of documents returned. For a query where only the intended target document was retrieved, the cell would contain "0/1". If the document could not be retrieved in a specific test, "NaN" is reported. For monolingual queries, no extra weighting information is available. Therefore, the gold standard queries all report "−" for the engine using balanced queries. Even using the gold standard queries, the engines return a different number of documents. This is caused by minor implementation differences between PostgreSQL and Xapian, such as using different stopword lists.

The summary table contains results for all documents used in these examples. Numbers in each cell are the median of results across all engines for that configuration/document combination. Any retrieval that fails to retrieve the intended document is given a score of 30, which is the worst score possible. The final column shows an average of these medians, and gives an indication of the overall performance of that configuration — low numbers indicate high precision.

Table 4.3: Summary of results, translations from Spanish to Norwegian

Config	D_1	D_2	D_3	D_4	D_5	D_6	D_7	**Average**
`gold-no`	0/18	0/22	0/25	0/31	0/31	0/26	0/28	0.0/25.86
`red1`	1/20.5	0/29.5	0/27.5	1/30.5	1/31	2/20	0/29.5	0.71/26.93
`red2`	3.5/28.5	1/29.5	4.5/27.5	1/30.5	1/31	4/21	1/29.5	2.29/28.21
`red3`	5/29.5	0.5/29.5	11/29	6/31	1/31	5/21	4/30	4.64/28.71
`red4`	6/29.5	2.5/29.5	10.5/30.5	6/31	1/31	6.5/24	4/30	5.21/29.36
`nolemtz`	1/20.5	0/24	1/27	1/30.5	1.5/30.5	2/20	0/29.5	0.93/26.0
`reverse`	14/27.5	1/24	1/21	5/31	3.5/31	2.5/23	1/29.5	4.0/26.71
`webdict`	5/25	0/28.5	14/27.5	7/30.5	0.5/31	9.5/22	4.5/25.5	5.79/27.14
`wnes-red1`	0/24.5	0/27.5	0/27.5	2/30.5	0/31	0.5/22	0/29.5	**0.36/27.5** [1]
`wnes-red2`	5/28.5	0/29.5	4.5/27.5	5/30.5	0.5/31	3.5/22	2.5/30	3.0/28.43
`wnes-red3`	7/29.5	0.5/29.5	11/29	6/31	0.5/31	3.5/22	4/30	4.64/28.86
`wnes-red4`	9.5/29.5	3/29.5	10/30.5	5/31	0.5/31	6.5/24	5.5/30	5.71/29.36
`wnno-red1`	0.5/29.5	0/27.5	9.5/30.5	11/25.5	0/31	4/30	3/29.5	4.0/29.07
`wnno-red2`	4/29.5	0.5/29.5	10.5/30.5	9.5/30	0/31	10/30	14.5/29.5	7.0/30.0
`wnno-red3`	9.5/30.5	0.5/29.5	12/30.5	10/30	0/31	11/30	13.5/30	8.07/30.21
`wnno-red4`	12/30.5	2/29.5	7.5/30.5	12.5/30	0.5/31	13/30	14.5/30	8.86/30.21
`wndb-red1`	0.5/29.5	0/27.5	9.5/30.5	14/26.5	1/31	4/30	3/29.5	4.57/29.21
`wndb-red2`	4/29.5	0/29.5	10.5/30.5	11/30	0/31	10/30	22/30	8.21/30.07
`wndb-red3`	10/30.5	0/29.5	12/30.5	12/30	0/31	13/30	13.5/30	8.64/30.21
`wndb-red4`	11.5/30.5	0/29.5	7.5/30.5	12.5/31	0/31	12/30	15/30	8.36/30.36
`wndb-reverse`	14/27.5	1.5/24	1/21	8/31	3.5/31	6/23	0/29.5	4.86/26.71
`wndb-webdict`	26.5/30	0/23.5	18/28.5	10.5/22.5	0.5/31	22/30	24/28	14.5/27.64

[1] Best average result obtained

Table 4.4: Summary of results, translations from Norwegian to Spanish

Config	D_1	D_2	D_3	D_4	D_5	D_6	D_7	**Average**
`gold-es`	0/30	0/28	0/23	0/31	0/31	0/30	0/21	0.0/27.71
`red1`	0/16	0.5/29	3/24	1/30	0/31	0/26	1/21	**0.79/25.29** [1]
`red2`	0/16	3/31	14/30.5	8.5/31	0/31	0/29	3.5/30	4.14/28.36
`red3`	0/16	5/31	13/30.5	10/31	1.5/31	0/30	2/30	4.5/28.5
`red4`	0/16	7/31	3.5/30.5	10/31	2/31	0.5/30	2.5/30.5	3.64/28.57
`nolemtz`	0/16	0/26	17.5/9.5 [3]	1/30	4/31	0/25	2/21	3.5/22.64
`reverse`	0/16	0/21	30/2 [3]	3/31	5/31	0/25	0/11	5.43/19.57
`webdict`	0/16	2.5/26	22/28.5	4.5/30	7.5/31	6/25	1/21	6.21/25.36
`wnes-red1`	0/16	0/28	17/21	1/30	0/31	0.5/28	30/20 [3]	6.93/24.86
`wnes-red2`	7.5/18	6/30	16/25	4/31	0/31	1/29	3.5/24	5.43/26.86
`wnes-red3`	7.5/18	11/30	15.5/31	7/31	1/31	3.5/29	2/29	6.79/28.43
`wnes-red4`	7.5/18	14/30	16.5/31	8/31	2.5/31	1/30	1.5/30	7.29/28.71
`wnno-red1`	0/16	0.5/29	**9/24** [2]	1/31	0/31	0/26	2/21	1.79/25.43
`wnno-red2`	0/16	3/31	16.5/30	12/31	0/31	0/29	3.5/30	5.00/28.29
`wnno-red3`	0/16	5/31	14/30	13.5/31	1/31	2.5/30	2/30	5.43/28.43
`wnno-red4`	0/16	7/31	8/30.5	15/31	3/31	4.5/30	2.5/30.5	5.71/28.57
`wndb-red1`	0/16	0/28	18/22	0/31	0/31	0.5/28	30/20 [3]	6.93/25.14
`wndb-red2`	7.5/18	6/30	16/24	1.5/31	0/31	1/29	3.5/24	5.07/26.71
`wndb-red3`	7.5/18	11/30	17/31	7/31	0.5/31	6/29	2/29	7.29/28.43
`wndb-red4`	7.5/18	14/30	17/31	12/31	3/30.5	3/30	1.5/30	8.29/28.64
`wndb-reverse`	0/16	0/21	30/2 [3]	3/31	6/31	1/25	0/11	5.71/19.57
`wndb-webdict`	0/16	1/25	18/26	2/30	2/31	20.5/24	7.5/20	7.29/24.57

[1] Best average result obtained
[2] Outlier on otherwise well-performing configuration
[3] Results with penalties for failing to retrieve the target document

4.3.2 Multi-word structures and compound words

Untranslated compound words

When translating from Norwegian, as covered in Section 2.1.1, compound words must be processed. The need for this is easily shown by an example using the queries written for document D_1, Example 4.1.

(4.1)

doktor	*folkefiende*	*helseskadelig*	*samfunnsdebatt.*
doctor	enemigo del pueblo	insalubre	debate social.
doctor	enemy of the people	detrimental to health	social debate.

When translating this query with the current implementation of CLIRch, only the word *doktor* ("doctor") is translated. The remaining terms are simply passed through unaltered. For each lexeme in this query, all the stems are covered by the dictionary. This means that correct compound processing could yield a gold standard query. A discussion on how this could be implemented in CLIRch can be found in Section 5.1.1.

As seen in Table A.1, the gold standard translation fares perfectly in engines `Pg-cd` and `Xap+w`. Effectively, for all engines and configurations, only the term "doctor" is used to search. In this case, this actually gives decent results — that single remaining query term is very relevant to the desired document.

Phrases mappable to compound words

In the opposite case, a phrase in Spanish can yield a series of short search terms in Norwegian where a compound word would be a more correct translation. This can be a serious hindrance to search performance. Again, the query defined for document D_1, Example 4.1, is a good example.

This query is expanded into a large amount of terms by all configurations, and the terms found are not incorrect. A large amount of documents are returned, however, and the retrieval result for PostgreSQL based search suffers. As the terms are quite generic, false positives are incurred.

Multi-word expressions that exist in the dictionary

(4.2) *sove ute* *gå seg vill* *møte andre mennesker*
dormir al aire libre perderse encontrar gente
sleep outside get lost meet other people
natur og byer *drive gatelangs* *i* *Paris og Istanbul*
naturaleza y ciudades vagar callejear Paris y Estambul
nature and cities wander the streets in Paris and Istanbul
krysse broer og grenser gå inn i fremmede land
cruzar puentes y fronteras entrar países extraños
cross bridges and borders enter strange countries
ukjente områder.
territorios desconocidos.
unknown areas.

Example 4.2, written for document D_5, uses several longer phrases. Many of these phrases are commonly used in either language, and thus likely to exist in a dictionary. A good example is the Norwegian phrase *gå seg vill*, which can be translated to Spanish as *perderse*. As the current CLIRch configuration processes each word without context, each word is translated to a potentially unrelated single term or set thereof.

In the example, good results are still obtained due to a high amount of remaining terms being correctly translated. This can be observed in Table A.5.

4.3.3 Dictionary and vocabulary related issues

Out-of-vocabulary words

Out-of-vocabulary words can severely reduce the quality of a query translation. Intuitively, a more seldom used word, one not found in a normal dictionary, is more likely to be a salient query constituent. Document D_6, with its queries as defined in Example 4.3, provides a good example of how important dictionary coverage is. Using the default engine, the only untranslated term is *livsnyter*, a compound term translatable to "enjoyer of life".

(4.3)

nyliberalisme	*uansvarlig*	*livsnyter*	*politikk*	*frihet*	*bygda*
neoliberalismo	irresponsable	vitalista	política	libertad	pueblo
neo-liberalism	irresponsible	libertine	politics	liberty	village

byen.
ciudad.
city.

In any configuration using the web-based dictionary, e.g. `webdict`, *nyliberalisme* and *uansvarlig* are also left untranslated. The impact is shown in Table A.6. All configurations using the web-based dictionary have a relatively high number of false positives.

Wrong choices when choosing between synonyms

When translating each term, the system often has to prune some results. The models of selection are not very complex in CLIRch, but they nevertheless fare acceptably. In the case of document D_7, however, there are three query words and two of those are translated poorly from Spanish to Norwegian. The translations selected are not incorrect, but do not match the vocabulary used in the text.

(4.4)

sensur	*samtid*	*aktuell.*
censura	contemporáneo	vigente.
censorship	contemporary	of current interest.

As can be observed in Table A.7, the reverse pruner does well for this example. This is not the case for most of the tests — it must therefore be assumed that reverse dictionary coverage overlaps well with the more relevant query term translation options for this specific query.

Named entities

In CLIRch, there is limited named entity recognition. If a named entity is detected by the lemmatizer, it is passed as a phrase to the following modules in the pipeline. For example, *Nueva York* would be detected as a Spanish named entity and passed to the dictionary as *nueva york*. In the example of

our current dictionary and language combination, this correctly yields *New York*. If an entity is not recognized, it is split and each word is translated separately.

In the setup currently used in CLIRch, named entity translation depends on dictionary coverage and relevant training data for the lemmatizers. Finding examples of either of these coming up short would be trivial in the current implementation of CLIRch.

4.3.4 Retaining grammatical information

Words that require lemmatization to be translated

To exemplify the need for lemmatization, document D_3 will be used. The query shown in Example 4.5 contains many inflected terms in both languages. This immediately effects configurations with no lemmatization and configurations using the reverse pruner, as shown in Table A.3.

(4.5)

høsten	*starter*	*skolen*	*tantene*	*toppluer*	*tenner*	*pupper*
el otoño	empiece	la escuela	tías	gorros	dientes	pechos
autumn	starts	school	aunts	caps	teeth	breasts

rullatorer	*gebiss*
andadores	dentaduras postizas
walking chairs	false teeth

For translation into Norwegian, only one term is correctly translated — "gebiss". This is incidentally enough to give decent results for this document. For translation into Spanish, results are heavily affected. Fewer documents are retrieved, and in several cases the target document is not returned at all.

Words that lose part of their original meaning in lemmatization

Using the same example as before, Example 4.2, a possible downside to lemmatization can be observed. The FreeLing lemmatizer reduces *perderse* (lose oneself) to its nonreflexive counterpart *perder* (lose). As detailed in Section 3.2.3, this is handled correctly in CLIRch through backoff.

Chapter 5

Future work and Conclusions

5.1 Future work

The following subsections contain discussions on the problems in the current implementation of CLIRch and suggestions on how these may be amended. Focus will be maintained on the selected language pair, Norwegian/Spanish, while other languages will be covered in Section 5.1.6.

5.1.1 Compound word splitting

The current implementation of CLIRch contains no module for splitting compound words. The compound word splitter implemented in [Ranang, 2010] is built on the NLTK [Bird et al., 2009] in Python — the same toolkit as CLIRch builds on. This means that reimplementing the compound word splitter as a CLIRch module is feasible, even though the code may need to be updated to conform to the newest version of NLTK.

If this proves difficult, other modules for performing compound word analysis exist. PostgreSQL includes a function that can perform decompounding given a properly formatted and tagged dictionary file [PostgreSQL, 2008]. For the example given in the earlier discussions, Example 4.1, the PostgreSQL implementation performed acceptably:

(5.1)

doktor	*folkefiende*	*helseskadelig*	*samfunnsdebatt.*
doktor	folk & fiende	helse & skade	samfunn & debatt.
doctor	people & enemy	health & damage	debate & society.

While testing this feature directly using SQL queries, it was shown to be somewhat unreliable for a range of terms, including "fiske" (fish, or fishing). E.g. "fiskehandel" and "torskefiske" were not split correctly. This may have been caused by a hastily compiled dictionary or other minor technical mistake. If reimplemented in Python, and coupled with an appropriate dictionary, it might produce decent results. This process would, however, require quite an amount of work.

5.1.2 Multi-word expression translation

A common occurrence in natural language is that of expressions holding figurative meaning, varyingly common expressions that cannot be correctly translated word by word. A good example is the commonly used Norwegian expression "i dag". A word-by-word translation yields "in day" whilst it unambiguously refers to "today".

To translate such terms, several approaches are possible. Alternative CLIR approaches using e.g. statistical machine translation may yield good results for common phrases. For a less holistic query translation system such as CLIRch, any bilingual term list containing complex phrases may be used. To be able to match such phrases, stemming or lemmatization may be needed on both the dictionary and the query. A backoff process, e.g. starting with testing the entire query and using smaller and smaller n-gram subsets, might also be needed.

5.1.3 Named entities

For supporting more named entities, a possible approach is using Wikipedia. A larger and larger amount of languages have a sizeable number of Wikipedia articles, and these articles are generally well-linked with articles in other languages. Proper nouns, titles and many other words are covered in Wikipedia, and there are constant updates to the articles and their connections.

Using Wikipedia as a dictionary can give some incorrect translations, however. For example, the article "Norges Konge" (The King of Norway) is connected to the English article "Monarchy of Norway". It may be advisable to run some sort of cleanup or relevance control, automatic or manual. It could also be used as a fallback after having failed to translate using a different dictionary.

5.1.4 Better utilization of dictionary metadata

Common expressions and other features in language may also be handled using a high quality machine readable dictionary (MRD), such as the proprietary one used in these experiments. The dictionary contains a wide range of phrase translations that were not included in the candidate translations used, mainly because phrase translation support was not implemented. If these lookups were implemented, translations of phrases such as the problematic "gå seg vill" from Section 4.3.2 could be produced.

The MRD also contains rich grammatical information and other metadata. Of particular use to Norwegian translation, it contains translations for partial compounds. Using the example from Section 4.3.2, the Spanish expression "debate social" is best translated into "samfunnsdebatt". This is covered in the Spanish-Norwegian dictionary by the translation of "social" into "samfunns-".

5.1.5 Lemmatization backoff granularity

In some cases, a more granular lemmatization backoff could be useful when looking up in the MRD. Some Spanish words can be used in both a masculine and a feminine form, e.g. "tío" and "tía", meaning uncle and aunt. When the plural form of aunt, "tías", is input to the system, it cannot be found in the MRD without lemmatization. The current Spanish lemmatizer, however, returns the masculine lemma "tío". This could be handled by adding specific rules to the backoff algorithm, or possibly by using the MRD. In the MRD, there are indications to the gender of words and whether or not they have alternate forms (e.g. may be written in feminine).

The example used only covers differences in gender. There are other examples in CLIRch where the lacking granularity in Spanish lemmatization

causes unnecessary loss of salient information. Reflexive verbs are stripped of their reflexivity, which can induce erroneous disambiguations. A simple example would be the Spanish phrase "se fueron", meaning "they left". This is stripped of its reflexivity, leaving "fueron". The translation finally produces the Norwegian word "være", "to be". This is the most likely interpretation of "fueron" but not a valid candidate translation at all for "se fueron".

5.1.6 Improving language support

Improvements for existing languages

Section 5.1 contains many specific measures that could improve retrieval efficiency in the current implementation of CLIRch. Especially for Norwegian retrieval, obtaining a more complete corpus would aid further work. A large corpus with a set of queries and corresponding relevant documents would give a better basis for evaluation.

A streamlined process for running a large set of benchmarks and producing easily comparable result reports would also be desirable. Using (or reimplementing functionality from) a standard tool for ad hoc retrieval evaluation might be beneficial — tools such as `trec_eval`[1] have already been created for exactly this purpose in relation to text retrieval conferences.

Adding more languages

Due to the differences in written language around the world, many languages might not see good results without creating certain language specific modules. Lemmatization is an obvious example of a module that should preferably be written to be language specific. For languages with no such resource available, approaches using statistical lemmatizers [Loponen & Järvelin, 2010] may be a good option.

Some languages have completely different word segmentation than the ones discussed thus far. Some languages are unsegmented, having no word boundary indication, such as Chinese or Japanese. Other languages have more complex morphology, such as Arabic. The measures used in [Levow et al., 2005]

[1] `http://trec.nist.gov/trec_eval/`

proved efficient for Chinese, Arabic, German, French and English, and implementing them as CLIRch modules would ease implementing support for many languages.

As mentioned in Section 2.1.2, approaches exist for extracting bilingual term lists from comparable corpora. Implementing such techniques in CLIRch would allow usage without having access to an MRD — obtaining such data can be one of the harder parts of building a CLIR pipeline for many languages. Using Wikipedia as a comparable corpus could quickly allow building pipelines for many languages.

5.2 Conclusions

Overall, CLIRch works as expected for the selected language pair. As can be seen in Table 4.3 and Table 4.4, the simple pipeline using 1-reduction pruning, `red1`, works well for both languages. Query side synonym expansion also appears to work well, especially when translating from Spanish to Norwegian. The Xapian integration achieves balanced queries, which improves performance when using multiple search translations — albeit not as much as using 1-reduction pruning.

Integration was implemented for document retrieval using both PostgreSQL and Xapian, and both gave decent retrieval results. The modular structure of CLIRch allows site specific addition and customization, as well as easing the process of managing contributed code. If it is adopted for use by students, academics or other interested parties, it will allow quick startup and minimize the need for reinventing the wheel — base components, as well as code for combining them, are already supplied.

If a readily available set of of dictionaries and corpora could be obtained, this would be of great benefit. Streamlining an integrated evaluation process for comparing configurations would also reduce the amount of work needed to start a CLIR project. Some additional components, such as a decompounding implementation for Norwegian or tokenizers for asian languages, would also increase the immediate benefits of adopting the framework. Even without these components, however, CLIRch can be considered a great asset.

Bibliography

[Aljlayl et al., 2002] Aljlayl, M., Frieder, O., & Grossman, D. (2002). On bidirectional English–Arabic search. *Journal of the American Society for Information Science and Technology*, 53(13), 1139–1151. Cited on pages 8 and 15.

[Bird et al., 2009] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media. Cited on pages 16 and 33.

[Cancedda et al., 2003] Cancedda, N., Déjean, H., Gaussier, É., Renders, J.-M., & Vinokourov, A. (2003). Report on CLEF-2003 Experiments: Two Ways of Extracting Multilingual Resources from Corpora. In *Comparative Evaluation of Multilingual Information Access Systems, 4th Workshop of the Cross-Language Evaluation Forum, CLEF 2003, Trondheim, Norway, August 21–22, 2003, Revised Selected Papers* (pp. 98–107). Cited on page 8.

[Clarke et al., 2000] Clarke, C. L., Cormack, G. V., & Tudhope, E. A. (2000). Relevance ranking for one to three term queries. *Information Processing and Management*, 36(2), 291 – 311. Cited on page 17.

[Demner-Fushman & Oard, 2003] Demner-Fushman, D. & Oard, D. W. (2003). The Effect of Bilingual Term List Size on Dictionary-Based Cross-Language Information Retrieval. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 4 - Volume 4*, HICSS '03 (pp. 108). Washington, DC, USA: IEEE Computer Society. Cited on page 7.

[Dumais et al., 1997] Dumais, S. T., Letsche, T. A., Littman, M. L., & Landauer, T. K. (1997). Automatic Cross-Language Retrieval Using Latent Semantic Indexing. In *Proc. of AAAI Symposium on Cross-Language Text and Speech Retrieval* (pp. 18–24). Cited on page 12.

[Federico & Bertoldi, 2002] Federico, M. & Bertoldi, N. (2002). Statistical Cross-Language Information Retrieval using N-Best Query Translations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02 (pp. 167–174). New York, NY, USA: ACM. Cited on page 8.

[Fellbaum & Vossen, 2007] Fellbaum, C. & Vossen, P. (2007). Connecting the universal to the specific: towards the global grid. In *Proceedings of the 1st international conference on Intercultural collaboration*, IWIC'07 (pp. 1–16). Berlin, Heidelberg: Springer-Verlag. Cited on page 8.

[Fjeld & Nygaard, 2009] Fjeld, R. V. & Nygaard, L. (2009). NorNet — a monolingual wordnet of modern Norwegian. In *NODALIDA 2009 workshop: WordNets and other Lexical Semantic Resources — between Lexical Semantics, Lexicography, Terminology and Formal Ontologies*, NEALT Proceedings Series Vol. 7 (pp. 13–16). Cited on page 22.

[Forner et al., 2011] Forner, P., Gonzalo, J., Kekäläinen, J., Lalmas, M., & De Rijke, M. (2011). *Multilingual and Multimodal Information Access Evaluation: Second International Conference of the Cross-Language Evaluation Forum, CLEF 2011, Amsterdam, The Netherlands, September 19-22, 2011, Proceedings.* Lecture Notes in Computer Science. Springer. Cited on page 4.

[Gollins & Sanderson, 2001] Gollins, T. & Sanderson, M. (2001). Improving cross language retrieval with triangulated translation. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*, SIGIR '01 (pp. 90–95). New York, NY, USA: ACM. Cited on page 8.

[Grossman & Frieder, 2004] Grossman, D. A. & Frieder, O. (2004). *Information retrieval: algorithms and heuristics.* Kluwer international series on information retrieval. Springer. Cited on page 5.

[Johannessen et al., 2011] Johannessen, J. B., Hagen, K., Lynum, A., & Nøklestad, A. (2011). OBT+stat: A combined rule-based and statistical tagger. In *Andersen, Gisle (ed.): Exploring Newspaper Language* Amsterdam: John Benjamins. Cited on page 17.

[Johannessen & Hauglin, 1996] Johannessen, J. B. & Hauglin, H. (1996). An automatic analysis of Norwegian compounds. In *Papers from the 16th Scandinavian Conference of Linguistics. Turku.* Cited on page 7.

[Keskustalo et al., 2002] Keskustalo, H., Hedlund, T., & Airio, E. (2002). UTACLIR : general query translation framework for several language pairs. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02 (pp. 448–448). New York, NY, USA: ACM. Cited on page 3.

[Levow & Oard, 2002] Levow, G.-A. & Oard, D. W. (2002). Signal boosting for translingual topic tracking: document expansion and n-best translation. In J. Allan (Ed.), *Topic Detection and Tracking: Event-based Information Organization* chapter 9, (pp. 175–195). Norwell, MA, USA: Kluwer Academic Publishers. Cited on pages 8 and 12.

[Levow et al., 2005] Levow, G.-A., Oard, D. W., & Resnik, P. (2005). Dictionary-based techniques for cross-language information retrieval. *Information Processing and Management*, 41(3), 523–547. Cited on pages 7, 10, 20, and 36.

[Loponen & Järvelin, 2010] Loponen, A. & Järvelin, K. (2010). A dictionary- and corpus-independent statistical lemmatizer for information retrieval in low resource languages. In *Proceedings of the 2010 international conference on Multilingual and multimodal information access evaluation: cross-language evaluation forum*, CLEF'10 (pp. 3–14). Berlin, Heidelberg: Springer-Verlag. Cited on page 36.

[Miniwatts Marketing Group, 2010] Miniwatts Marketing Group (2010). Internet world users by language. `http://www.internetworldstats.com/stats7.htm`. Cited on page 1.

[Oard et al., 2000] Oard, D. W., Levow, G.-A., & Cabezas, C. I. (2000). CLEF experiments at the University of Maryland: Statistical stemming and backoff translation strategies. In *Working Notes of the First Cross-Language Evaluation Forum (CLEF-1)*: Springer. Cited on page 7.

[Padró et al., 2010] Padró, L., Collado, M., Reese, S., Lloberes, M., & Castellón, I. (2010). FreeLing 2.1: Five Years of Open-source Language Processing Tools. In N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, & D. Tapias (Eds.), *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)* Valletta, Malta: European Language Resources Association (ELRA). Cited on pages 17 and 22.

[Pirkola, 1998] Pirkola, A. (1998). The Effects of Query Structure and Dictionary-Setups in Dictionary-Based Cross-language Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR*

Conference on Research and Development in Information Retrieval (pp. 55–63). Cited on page 12.

[Porter, 1980] Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130–137. Cited on page 6.

[PostgreSQL, 2008] PostgreSQL (2008). PostgreSQL 8.3 Documentation: Full Text Search Dictionaries. `http://www.postgresql.org/docs/8.3/static/textsearch-dictionaries.html`. Cited on pages 17 and 33.

[Q-Success, 2012] Q-Success (2012). W3techs monthly survey: Usage of content languages for websites. `http://w3techs.com/technologies/overview/content_language/all`. Cited on page 1.

[Ranang, 2010] Ranang, M. T. (2010). *Open-Domain Word-Level Interpretation of Norwegian : Towards a General Encyclopedic Question-Answering System for Norwegian.* Doctoral thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. Cited on pages 7 and 33.

[Robertson & Zaragoza, 2009] Robertson, S. E. & Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4), 333–389. Cited on page 9.

[Sahlgren & Karlgren, 2002] Sahlgren, M. & Karlgren, J. (2002). Vector-Based Semantic Analysis Using Random Indexing for Cross-Lingual Query Expansion. In *Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, CLEF '01 (pp. 169–176). London, UK, UK: Springer-Verlag. Cited on page 12.

[Salton et al., 1975] Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), 613–620. Cited on page 9.

[Singhal & Pereira, 1999] Singhal, A. & Pereira, F. (1999). Document Expansion for Speech Retrieval. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99 (pp. 34–41). New York, NY, USA: ACM. Cited on page 10.

[Spärck Jones, 1972] Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28, 11–21. Cited on page 9.

[Yang & Kirchhoff, 2006] Yang, M. & Kirchhoff, K. (2006). Phrase-Based Backoff Models for Machine Translation of Highly Inflected Languages. In *Proceedings of the 21st International Conference on Computational Linguistics* (pp. 1017–1020). Cited on page 7.

Appendix A

Result tables

Table A.1: Results for queries searching for text D_1

Translations from Spanish to Norwegian

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-no	0/18	0/18	0/17	–
red1	2/21	4/21	0/20	0/20
red2	7/29	8/29	0/28	0/28
red3	13/30	9/30	1/29	0/29
red4	11/30	12/30	1/29	1/29
nolemtz	2/21	4/21	0/20	0/20
reverse	13/28	13/28	15/27	17/27
webdict	1/25	10/25	5/25	5/25
wnes-red1	0/25	8/25	0/24	0/24
wnes-red2	10/29	14/29	0/28	0/28
wnes-red3	15/30	14/30	0/29	0/29
wnes-red4	16/30	15/30	4/29	3/29
wnno-red1	1/30	20/30	0/29	0/29
wnno-red2	7/30	21/30	1/29	1/29
wnno-red3	17/31	21/31	2/30	1/30
wnno-red4	19/31	21/31	5/30	1/30
wndb-red1	1/30	20/30	0/29	0/29
wndb-red2	7/30	21/30	1/29	1/29
wndb-red3	17/31	21/31	3/30	1/30
wndb-red4	18/31	21/31	5/30	3/30
wndb-reverse	14/28	13/28	14/27	17/27
wndb-webdict	28/30	27/30	26/30	26/30

Translations from Norwegian to Spanish

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-es	0/30	2/30	0/31	–
red1	1/16	0/16	0/16	0/16
red2	1/16	0/16	0/16	0/16
red3	1/16	0/16	0/16	0/16
red4	1/16	0/16	0/16	0/16
nolemtz	1/16	0/16	0/16	0/16
reverse	1/16	0/16	0/16	0/16
webdict	1/16	0/16	0/16	0/16
wnes-red1	1/16	0/16	0/16	0/16
wnes-red2	6/18	0/18	9/18	9/18
wnes-red3	6/18	0/18	9/18	9/18
wnes-red4	6/18	0/18	9/18	9/18
wnno-red1	1/16	0/16	0/16	0/16
wnno-red2	1/16	0/16	0/16	0/16
wnno-red3	1/16	0/16	0/16	0/16
wnno-red4	1/16	0/16	0/16	0/16
wndb-red1	1/16	0/16	0/16	0/16
wndb-red2	6/18	0/18	9/18	9/18
wndb-red3	6/18	0/18	9/18	9/18
wndb-red4	6/18	0/18	9/18	9/18
wndb-reverse	1/16	0/16	0/16	0/16
wndb-webdict	1/16	0/16	0/16	0/16

Table A.2: Results for queries searching for text D_2

Translations from Spanish to Norwegian

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-no	0/22	0/22	0/21	–
red1	0/30	8/30	0/29	0/29
red2	2/30	20/30	0/29	0/29
red3	1/30	20/30	0/29	0/29
red4	5/30	20/30	0/29	0/29
nolemtz	0/24	3/24	0/24	0/24
reverse	2/24	7/24	0/24	0/24
webdict	0/29	5/29	0/28	0/28
wnes-red1	0/28	5/28	0/27	0/27
wnes-red2	0/30	19/30	0/29	0/29
wnes-red3	1/30	20/30	0/29	0/29
wnes-red4	6/30	20/30	0/29	0/29
wnno-red1	0/28	5/28	0/27	0/27
wnno-red2	1/30	19/30	0/29	0/29
wnno-red3	1/30	19/30	0/29	0/29
wnno-red4	3/30	19/30	1/29	0/29
wndb-red1	0/28	5/28	0/27	0/27
wndb-red2	0/30	19/30	0/29	0/29
wndb-red3	0/30	12/30	0/29	0/29
wndb-red4	0/30	15/30	0/29	0/29
wndb-reverse	2/24	7/24	1/24	0/24
wndb-webdict	0/24	2/24	0/23	0/23

Translations from Norwegian to Spanish

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-es	0/28	0/28	0/31	–
red1	1/29	2/29	0/29	0/29
red2	10/31	6/31	0/31	0/31
red3	12/31	9/31	1/31	0/31
red4	14/31	13/31	1/31	0/31
nolemtz	0/26	4/26	0/26	0/26
reverse	0/21	3/21	0/21	0/21
webdict	3/26	10/26	2/26	2/26
wnes-red1	0/28	6/28	0/28	0/28
wnes-red2	11/30	13/30	1/30	1/30
wnes-red3	14/30	15/30	8/30	1/30
wnes-red4	17/30	16/30	12/30	2/30
wnno-red1	1/29	2/29	0/29	0/29
wnno-red2	10/31	6/31	0/31	0/31
wnno-red3	12/31	9/31	1/31	0/31
wnno-red4	14/31	13/31	1/31	0/31
wndb-red1	0/28	6/28	0/28	0/28
wndb-red2	11/30	13/30	1/30	1/30
wndb-red3	14/30	15/30	8/30	1/30
wndb-red4	17/30	16/30	12/30	2/30
wndb-reverse	0/21	3/21	0/21	0/21
wndb-webdict	1/25	2/25	1/25	1/25

Table A.3: Results for queries searching for text D_3

Translations from Spanish to Norwegian

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-no	0/25	1/25	0/25	–
red1	0/28	14/28	0/27	0/27
red2	3/28	15/28	6/27	3/27
red3	13/30	15/30	9/28	6/28
red4	11/31	15/31	10/30	6/30
nolemtz	0/27	8/27	1/27	1/27
reverse	0/21	4/21	1/21	1/21
webdict	8/28	13/28	15/27	15/27
wnes-red1	0/28	14/28	0/27	0/27
wnes-red2	3/28	15/28	6/27	3/27
wnes-red3	13/30	15/30	9/28	6/28
wnes-red4	11/31	15/31	9/30	5/30
wnno-red1	14/31	17/31	5/30	5/30
wnno-red2	14/31	17/31	7/30	5/30
wnno-red3	15/31	18/31	9/30	6/30
wnno-red4	13/31	17/31	2/30	2/30
wndb-red1	14/31	17/31	5/30	5/30
wndb-red2	14/31	17/31	7/30	5/30
wndb-red3	15/31	18/31	9/30	6/30
wndb-red4	13/31	17/31	2/30	2/30
wndb-reverse	0/21	4/21	1/21	1/21
wndb-webdict	18/29	16/29	18/28	18/28

Translations from Norwegian to Spanish

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-es	0/23	1/23	0/31	–
red1	0/24	1/24	5/24	5/24
red2	1/30	14/30	14/31	14/31
red3	1/30	12/30	14/31	14/31
red4	1/30	9/30	3/31	4/31
nolemtz	0/12	5/12	NaN/7	NaN/7
reverse	NaN/2	NaN/2	NaN/2	NaN/2
webdict	12/26	18/26	26/31	26/31
wnes-red1	3/21	15/21	19/21	19/21
wnes-red2	2/25	16/25	16/25	16/25
wnes-red3	1/31	18/31	15/31	16/31
wnes-red4	12/31	18/31	16/31	17/31
wnno-red1	0/24	12/24	9/24	9/24
wnno-red2	2/30	16/30	17/30	17/30
wnno-red3	1/30	15/30	13/30	16/30
wnno-red4	2/30	15/30	6/31	10/31
wndb-red1	1/22	18/22	18/22	18/22
wndb-red2	9/24	15/24	17/24	18/24
wndb-red3	14/31	18/31	17/31	17/31
wndb-red4	15/31	18/31	17/31	17/31
wndb-reverse	NaN/2	NaN/2	NaN/2	NaN/2
wndb-webdict	12/26	18/26	18/26	18/26

Table A.4: Results for queries searching for text D_4

Translations from Spanish to Norwegian

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-no	0/31	4/31	0/30	–
red1	0/31	5/31	1/30	1/30
red2	0/31	5/31	1/30	1/30
red3	0/31	6/31	6/31	6/31
red4	0/31	6/31	6/31	6/31
nolemtz	0/31	5/31	1/30	1/30
reverse	0/31	9/31	5/31	5/31
webdict	0/31	9/31	7/30	7/30
wnes-red1	0/31	5/31	2/30	2/30
wnes-red2	0/31	11/31	5/30	5/30
wnes-red3	0/31	12/31	6/31	6/31
wnes-red4	0/31	12/31	5/31	5/31
wnno-red1	14/26	17/26	8/25	8/25
wnno-red2	0/30	12/30	10/30	9/30
wnno-red3	0/30	14/30	10/30	10/30
wnno-red4	0/30	14/30	11/30	14/30
wndb-red1	18/27	18/27	10/26	10/26
wndb-red2	1/30	17/30	11/30	11/30
wndb-red3	0/30	15/30	12/30	12/30
wndb-red4	0/31	15/31	12/31	13/31
wndb-reverse	0/31	9/31	10/31	7/31
wndb-webdict	14/23	20/23	7/22	7/22

Translations from Norwegian to Spanish

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-es	0/31	8/31	0/31	–
red1	0/30	8/30	1/30	1/30
red2	0/31	17/31	11/31	6/31
red3	0/31	15/31	14/31	6/31
red4	0/31	16/31	14/31	6/31
nolemtz	0/30	17/30	1/30	1/30
reverse	0/31	18/31	3/31	3/31
webdict	0/30	8/30	4/30	5/30
wnes-red1	0/30	13/30	1/30	1/30
wnes-red2	0/31	16/31	5/31	3/31
wnes-red3	0/31	18/31	11/31	3/31
wnes-red4	1/31	18/31	13/31	3/31
wnno-red1	0/31	13/31	1/31	1/31
wnno-red2	0/31	18/31	14/31	10/31
wnno-red3	0/31	18/31	15/31	12/31
wnno-red4	4/31	18/31	15/31	15/31
wndb-red1	0/31	18/31	0/31	0/31
wndb-red2	0/31	17/31	3/31	0/31
wndb-red3	1/31	18/31	11/31	3/31
wndb-red4	9/31	18/31	15/31	8/31
wndb-reverse	0/31	18/31	3/31	3/31
wndb-webdict	0/30	11/30	2/30	2/30

Table A.5: Results for queries searching for text D_5

Translations from Spanish to Norwegian

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-no	0/31	0/31	0/31	–
red1	0/31	0/31	2/31	2/31
red2	0/31	0/31	3/31	2/31
red3	0/31	0/31	5/31	2/31
red4	0/31	0/31	5/31	2/31
nolemtz	0/31	0/31	3/30	3/30
reverse	3/31	1/31	4/31	4/31
webdict	0/31	0/31	1/31	1/31
wnes-red1	0/31	0/31	0/31	0/31
wnes-red2	0/31	0/31	2/31	1/31
wnes-red3	0/31	0/31	3/31	1/31
wnes-red4	0/31	0/31	3/31	1/31
wnno-red1	0/31	0/31	0/31	0/31
wnno-red2	0/31	0/31	0/31	0/31
wnno-red3	0/31	0/31	0/31	0/31
wnno-red4	0/31	0/31	2/31	1/31
wndb-red1	0/31	0/31	2/31	2/31
wndb-red2	0/31	0/31	0/31	0/31
wndb-red3	0/31	0/31	0/31	0/31
wndb-red4	0/31	0/31	0/31	0/31
wndb-reverse	3/31	1/31	4/31	4/31
wndb-webdict	0/31	0/31	1/31	1/31

Translations from Norwegian to Spanish

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-es	0/31	0/31	0/31	–
red1	1/31	0/31	0/31	0/31
red2	1/31	0/31	0/31	0/31
red3	7/31	1/31	2/31	0/31
red4	4/31	1/31	3/31	1/31
nolemtz	2/31	0/31	6/31	7/31
reverse	5/31	0/31	5/31	8/31
webdict	0/31	0/31	15/31	15/31
wnes-red1	0/31	0/31	0/31	0/31
wnes-red2	0/31	0/31	0/31	0/31
wnes-red3	5/31	0/31	2/31	0/31
wnes-red4	8/31	1/31	4/31	1/31
wnno-red1	1/31	0/31	0/31	0/31
wnno-red2	1/31	0/31	0/31	0/31
wnno-red3	3/31	0/31	2/31	0/31
wnno-red4	4/31	2/31	4/31	1/31
wndb-red1	0/31	0/31	0/31	0/31
wndb-red2	0/31	0/31	0/31	0/31
wndb-red3	2/31	0/31	1/31	0/31
wndb-red4	12/30	4/30	2/31	0/31
wndb-reverse	5/31	0/31	8/31	7/31
wndb-webdict	1/31	1/31	3/31	4/31

Table A.6: Results for queries searching for text D_6

Translations from Spanish to Norwegian

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-no	0/26	4/26	0/25	–
red1	4/20	7/20	0/20	0/20
red2	6/21	12/21	1/21	2/21
red3	6/21	12/21	2/21	4/21
red4	8/24	15/24	3/24	5/24
nolemtz	4/20	7/20	0/20	0/20
reverse	4/23	13/23	1/23	1/23
webdict	11/22	16/22	8/22	8/22
wnes-red1	1/22	15/22	0/22	0/22
wnes-red2	6/22	16/22	1/22	1/22
wnes-red3	4/22	15/22	1/22	3/22
wnes-red4	8/24	15/24	3/24	5/24
wnno-red1	6/30	20/30	2/30	2/30
wnno-red2	16/30	21/30	4/30	4/30
wnno-red3	14/30	20/30	7/30	8/30
wnno-red4	14/30	20/30	8/30	12/30
wndb-red1	6/30	20/30	2/30	2/30
wndb-red2	16/30	21/30	4/30	4/30
wndb-red3	16/30	21/30	8/30	10/30
wndb-red4	14/30	20/30	8/30	10/30
wndb-reverse	4/23	13/23	8/23	1/23
wndb-webdict	24/30	21/30	22/30	22/30

Translations from Norwegian to Spanish

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-es	0/30	11/30	0/30	–
red1	0/26	5/26	0/26	0/26
red2	0/29	12/29	0/29	0/29
red3	0/30	13/30	0/30	0/30
red4	1/30	13/30	0/30	0/30
nolemtz	0/25	2/25	0/25	0/25
reverse	0/25	3/25	0/25	0/25
webdict	3/25	6/25	6/25	6/25
wnes-red1	1/28	11/28	0/28	0/28
wnes-red2	2/29	14/29	0/29	0/29
wnes-red3	7/29	11/29	0/29	0/29
wnes-red4	1/30	12/30	1/30	0/30
wnno-red1	0/26	5/26	0/26	0/26
wnno-red2	0/29	12/29	0/29	0/29
wnno-red3	5/30	13/30	0/30	0/30
wnno-red4	7/30	13/30	2/30	0/30
wndb-red1	1/28	11/28	0/28	0/28
wndb-red2	2/29	14/29	0/29	0/29
wndb-red3	12/29	18/29	0/29	0/29
wndb-red4	5/30	17/30	1/30	0/30
wndb-reverse	0/25	3/25	2/25	0/25
wndb-webdict	22/24	22/24	19/24	19/24

Table A.7: Results for queries searching for text D_7

Translations from Spanish to Norwegian

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-no	1/28	0/28	0/27	–
red1	0/30	2/30	0/29	0/29
red2	1/30	2/30	1/29	1/29
red3	11/30	5/30	2/30	3/30
red4	11/30	5/30	2/30	3/30
nolemtz	0/30	2/30	0/29	0/29
reverse	1/30	2/30	1/29	1/29
webdict	2/26	3/26	6/25	6/25
wnes-red1	0/30	2/30	0/29	0/29
wnes-red2	5/30	3/30	2/30	2/30
wnes-red3	11/30	5/30	2/30	3/30
wnes-red4	8/30	13/30	2/30	3/30
wnno-red1	2/30	3/30	3/29	3/29
wnno-red2	14/30	4/30	15/29	15/29
wnno-red3	14/30	7/30	14/30	13/30
wnno-red4	15/30	9/30	15/30	14/30
wndb-red1	2/30	3/30	3/29	3/29
wndb-red2	21/30	9/30	23/30	23/30
wndb-red3	14/30	7/30	14/30	13/30
wndb-red4	15/30	9/30	16/30	15/30
wndb-reverse	0/30	2/30	0/29	0/29
wndb-webdict	24/28	23/28	24/28	24/28

Translations from Norwegian to Spanish

Conf	Pg-trad	Pg-cd	Xap	Xap+w
gold-es	1/21	0/21	0/21	–
red1	8/21	1/21	1/21	1/21
red2	3/30	1/30	4/30	4/30
red3	2/30	1/30	2/30	2/30
red4	6/31	3/31	2/30	2/30
nolemtz	8/21	1/21	3/21	1/21
reverse	0/11	0/11	0/11	0/11
webdict	8/21	1/21	1/21	1/21
wnes-red1	NaN/20	NaN/20	NaN/20	NaN/20
wnes-red2	5/24	3/24	3/24	4/24
wnes-red3	2/29	3/29	1/29	2/29
wnes-red4	2/29	3/29	1/31	1/31
wnno-red1	8/21	1/21	3/21	1/21
wnno-red2	3/30	1/30	4/30	4/30
wnno-red3	2/30	1/30	2/30	2/30
wnno-red4	6/31	3/31	2/30	2/30
wndb-red1	NaN/20	NaN/20	NaN/20	NaN/20
wndb-red2	5/24	3/24	3/24	4/24
wndb-red3	2/29	3/29	1/29	2/29
wndb-red4	2/29	3/29	1/31	1/31
wndb-reverse	0/11	0/11	0/11	0/11
wndb-webdict	6/20	1/20	9/20	9/20