Linguistically-Adapted Structural Query Annotation for Digital Libraries in the Social Sciences

Caroline Brun **Vassilina Nikoulina** **Nikolaos Lagos**

Xerox Research Centre Europe
6, chemin de Maupertuis
38240, Meylan France
{firstname.lastname}@xrce.xerox.com

Abstract

Query processing is an essential part of a range of applications in the social sciences and cultural heritage domain. However, out-of-the-box natural language processing tools originally developed for full phrase analysis are inappropriate for query analysis. In this paper, we propose an approach to solving this problem by adapting a complete and integrated chain of NLP tools, to make it suitable for queries analysis. Using as a case study the automatic translation of queries posed to the Europeana library, we demonstrate that adapted linguistic processing can lead to improvements in translation quality.

1 Introduction

Query processing tools are essential components of digital libraries and content aggregators. Their operation varies from simple stop word removal and stemming to advanced parsing, that treats queries as a collection of phrases rather than single terms (Mothe and Tanguy, 2007). They are used in a range of applications, from information retrieval (via search engines that provide access to the digital collections) to query analysis.

Current query processing solutions tend to use out-of-the-box Natural Language Processing (NLP) tools that were originally developed for full phrase analysis, being inappropriate for query analysis.

Correct query annotation and interpretation is even more important in the cultural heritage or social sciences domain, as a lot of the content can be in multimedia form and only metadata (most of the times in the form of tags) is exploitable by traditional text-oriented information retrieval and analysis techniques.

Furthermore, as recent studies of user querying behavior mention, queries in these domains are not only very short but are also quite specific in terms of content: they refer to artist names, titles, dates, and objects (Koolen and Kamps, 2010; Ireson and Oomen, 2007). Take the example of a query like *"coupe apollon"* ("bowl apollon"). While in standard analysis *"coupe"* would be identified as a verb ("couper", i.e. "to cut"), in the context of a query it should be actually tagged as a noun, which refers to an object. Such a difference may lead to different preprocessing and worse retrieval.

In this paper, we propose an approach to solving this problem by adapting a complete and integrated chain of NLP tools, based on the Xerox Incremental Parser (XIP), to make it suitable for queries' analysis. The adaptation includes recapitalization, adapted Part of Speech (PoS) tagging, adapted chunking and Named Entities (NE) recognition. We claim that several heuristics especially important for queries' analysis, such as favoring nominal interpretations, result in improved linguistic structures, which can have an impact in a wide range of further applications (e.g. information retrieval, query translation, information extraction, query reformulation etc.).

2 Prior art

The problem of adapted query processing, often referred to as structural query annotation, includes capitalization, NEs detection, PoS tagging and query segmentation. Most of the existing works treat each of these steps independently and address only one of the above issues.

Many works address the problem of query segmentation. According to Tan and Peng (2008), query segmentation is a problem which is close to the chunking problem, but the chunking problem is directly linked to the PoS tagging results, which are often noisy for the queries. Thus, most of the works on query segmentation are based on the statistical interaction between a pair of query words to identify the border between the segments in the query (Jones et al., 2006; Guo et

Proceedings of the 6th EACL Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, pages 55–64,
Avignon, France, 24 April 2012. ©2012 Association for Computational Linguistics

al., 2008). Tan and Peng (2008) propose a generative language model enriched with Wikipedia to identify "concepts" rather than simply "frequency-based" patterns. The segmentation proposed by Bergsma and Wang (2007) is closer to the notion of NP chunking. They propose a machine-learned query segmentation system trained on manually annotated set of 500 AOL queries. However, in this work PoS tagging is used as one of the features in query segmentation and is done with a generic PoS tagger, non adapted for queries.

PoS tagging is an important part of query processing and used in many information analytics tasks (query reformulation, query segmentation, etc.). However very few works address query-oriented PoS tagging. Allan and Raghavan (2002) consider that PoS tagging might be ambiguous for short queries and propose to interact with the user for disambiguation. Barr et al. (2008) produce a set of manually annotated queries, and then train a Brill tagger on this set in order to create an adapted PoS tagger for search queries.

A notable work is the one by Bendersky et al. (2010), which addresses the capitalization, PoS tagging and query segmentation in the same paper. However, this approach proposes for each of the above steps a probabilistic model that relies on the document corpus rather on the query itself. Such an approach is not applicable for most digital content providers who would reluctantly give access to their document collection. Moreover, the query expansion, which is the central idea of the described approach, is not possible for most of digital libraries that are organized in a database. Secondly, Bendersky et al. (2010) proposes adapting each processing step independently. Although this is not mentioned in the paper, these three steps can be applied in a sequence, where PoS tagging can profit from the recapitalization, and chunking from the PoS tagging step. However, once the recapitalization is done, it can not be changed in the following steps. This work doesn't address the adaptation of the NE recognition component, as we do, and which might change the final chunking and PoS tagging in certain cases.

In our approach, part of the recapitalization is done during the PoS tagging, in interaction with the NE recognition, which allows us to consider these two steps as interleaved. Moreover, the linguistic processing we propose is generic: corpus-independent (at least most of its parts except for NE recognition) and doesn't require access to the document collection.

3 Data

This work is based on search logs from Europeana[1]. These are real users' queries, where Named Entities are often lowercased and the structures are very different from normal phrase structure. Thus, this data is well adapted to demonstrate the impact of adapted linguistic processing.

4 Motivation

We show the importance of the adapted linguistic query processing using as example the task of query translation, a real need for today's digital content providers operating in a multilingual environment. We took a sample of Europeana queries and translated them with different MT systems: in-house (purely statistical) or available online (rule-based). Some examples of problematic translations are shown in the Table 1.

	Input query	**Automatic Translation**	**Human translation**
	French-English		
1	journal panorama paris	newspaper panorama bets	newspaper panorama paris
2	saint jean de luz	saint jean of luz	saint jean de luz
3	vie et mort de l'image	life and died of the image	life and death of image
4	langue et réalité	and the reality of language	language and reality
	English-French		
5	maps europe	trace l'Europe	cartes de l'Europe
6	17th century saw	Du 17ème siècle a vu	scie du 17ème siècle
7	chopin george sand	george sable chopin soit	chopin george sand

Table 1: Examples of the problematic query translations

[1] A portal that acts as an interface to millions of digitized records, allowing users to explore Europe's cultural heritage. For more information please visit http://www.europeana.eu/portal/

Although in general, the errors done by statistical and rule-based models are pretty different, there are some common errors done in the case of the query translation. Both models, being designed for full-sentence translation, find the query structure very unnatural and tend to reproduce the full sentence in the output (ex. 1, 3, 4, 5, 6). The errors may come either from a wrong PoS tagging (for rule-based systems), or from the wrong word order (statistical-based systems), or from the choice of the wrong translation (both types of systems).

One might think that the word order problem is not crucial for queries, because most of the IR models use the bag of words models, which ignore the order of words. However, it might matter in some cases: for example, if *and/or* are interpreted as a logical operator, it is important to place them correctly in the sentence (examples3, 4).

Errors also may happen when translating NEs (ex.1, 2, 7). The case information, which is often missing in the real-life queries, helps to deal with the NEs translation.

The examples mentioned above illustrate that adapted query processing is important for a task such as query translation, both in the case of rule-based and empirical models. Although the empirical models can be adapted if an appropriately sized corpus exists, such a corpus is not always available.

Thus we propose adapting the linguistic processing prior to query translation (which is further integrated in the SMT model). We demonstrate the feasibility and impact of our approach based on the difference in translation quality but the adaptations can be useful in a number of other tasks involving query processing (e.g. question answering, query logs analysis, etc.).

5 Linguistic Processing Adaptation

As said before, queries have specific linguistic properties that make their analysis difficult for standard NLP tools. This section describes the approach we have designed to improve query chunking. Following a study of the corpus of query logs, we rely on the specific linguistic properties of the queries to adapt different steps of linguistic analysis, from preprocessing to chunking.

These adaptations consist in the following very general processes, for both English and French:

Recapitalization: we recapitalize, in a preprocessing step, some uncapitalized words in queries that can be proper nouns when they start with a capital letter.

Part of Speech disambiguation:

- the part of speech tagging favors nominal interpretation (whereas standard part of speech taggers are designed to find a verb in the input, as PoS tagging generally applies on complete sentences);
- the recapitalization information transmitted from the previous step is used to change the PoS interpretation in some contexts.

Chunking: the chunking is improved by:

- considering that a full NE is a chunk, which is not the case in standard text processing, where a NE can perfectly be just a part of a chunk;
- grouping coordinated NEs of the same type;
- performing PP and AP attachment with the closest antecedent that is morphologically compatible

These processes are very general and may apply to queries in different application domains, with maybe some domain-dependent adaptations (for example, NEs may change across domains).

These adaptations have been implemented within the XIP engine, for the French and English grammars. The XIP framework allows integrating the adaptations of different steps of query processing into a unified framework, where the changes from one step can influence the result of the next step: the information performed at a given step is transmitted to the next step by XIP through linguistic features.

5.1 Preprocessing

Queries are often written with misspelling errors, in particular for accents and capital letters of NEs. See the following query examples extracted from our query log corpus:

```
lafont Robert (French query)
henry de forge et jean mauclère
(French query)
muse prado madrid (French query)
carpaccio queen cornaro (English
query)
man ray (English query)
```

This might be quite a problem for linguistic treatments, like PoS tagging and of course NE recognition, which often use capital letter information as a triggering feature.

Recapitalizing these words at the preprocessing step of a linguistic analysis, i.e. during the morphological analysis, is technically relatively easy, however it would be an important generator of spurious ambiguities in the context of full sentence parsing (standard context of linguistic parsing). Indeed, considering that all lower case words that can be proper nouns with a capital letter should also have capitalized interpretation, such as *price, jean, read, us, bush, lay*, etc., in English or *pierre, médecin, …* in French) would be problematic for a PoS tagger as well as for a NE recognizer. That's why it is not performed in a standard analysis context, considering also that misspelling errors are not frequent in "standard" texts. In the case of queries however, they are frequent, and since queries are far shorter in average than full sentences the tagging can be adapted to this context (see next section), we can afford to perform recapitalization using the following methodology, combining lexical information and contextual rules:

1. The preprocessing lexicon integrates all words starting with a lower case letter which can be first name (*henry, jean, isaac …*), family and celebrity name (*chirac, picasso...*) and place names (*paris, saint pétersbourg, …*) when capitalized.

2. When an unknown word starting with a lower case letter is preceded by a first name and eventually by a particle (*de, van, von …*), it is analyzed as a last name, in order to be able to trigger standard NE recognition. This is one example of interleaving of the processes: here part-of-speech interpretation is conditioned by the recapitalization steps which transmits information about recapitalization (via features within XIP) that triggers query-specific pos disambiguation rules.

The recapitalization (1) has been implemented within the preprocessing components of XIP within finite state transducers (see (Karttunen, 2000)). The second point (2) is done directly within XIP in the part-of-speech tagging process, with a contextual rule. For example, the analysis of the input query "*jean mauclère*" gets the following structure and dependency output with the standard French grammar.

```
Query: jean mauclère
NMOD(jean, mauclère)
0>GROUP[NP[jean] AP[mauclère]]
```

Because *jean* is a common noun and mauclère is an unknown word which has been guessed as an adjective by the lexical guesser.

It gets the following analysis with the preprocessing adaptations described above:

```
NMOD(jean,mauclère)
PERSON_HUM(jean mauclère)
FIRSTNAME(jean,jean mauclère)
LASTNAME(mauclère,jean mauclère)
0>GROUP[NP[NOUN[jean mauclère]]]
```

Because *jean* has been recognized as a first name and consequently the unknown word after has been inferred has a proper noun (last name) by the pos tagging contextual rule; the recapitalization process and part-of-speech interpretation are therefore interleaved.

5.2 Part of speech disambiguation

In the context of query analysis, part-of-speech tagging has to be adapted also, since standard part-of-speech disambiguation strategies aim generally at disambiguating in the context of full sentences. But queries are very different from full sentences: they are mostly nominal with sometimes infinitive, past participial, or gerundive insertions, e.g.:

```
statuettes hommes jouant avec un
chien (French query)
coupe apollon (French query)
architecture    musique    (French
query)
statue haut relief grecque du 5
siecle (French query)
david playing harp fpr saul (Eng-
lish query)
stained   glass   angel   (English
query)
```

Standard techniques for part-of-speech tagging include rule based methods and statistical methods, mainly based on hidden Markov models (see for example (Chanod and Tapanainen, 1995)). In this case, it would be possible to recompute the probabilities on a corpus of queries manually annotated. However, the correction of part-of-speech tags in the context of queries is easy to develop with a small set of rules. We focus on English and French, and in queries, the main problems come from the ambiguity between noun and verbs, which has to be solved differently than in the context of a standard sentence.

The approach we adopt to correct the tagging with the main following contextual rules:

- If there is a noun/verb ambiguity:
    - If the ambiguity is on the first word of the query (e.g. "*coupe apollon*", "*oil flask*"), select the noun interpretation;
    - If the ambiguity is on the second word of the query, prefer the noun interpretation if the query starts with an adjective or a noun (e.g. in "*young people social competences*", select the noun interpretation for *people*, instead of verb)
    - Select noun interpretation if there is no person agreement with one of the previous nouns (e.g. "*les frères bissons*", *frères* belongs to the 3rd person but *bissons* to the 1st one of the verb "*bisser*")
    - For a verb which is neither at the past participle form nor the infinitive form, select the noun interpretation if it is not followed by a determiner (e.g. "*tremblement terre lisbonne*", *terre* is disambiguated as a noun"))
    - Choose the noun interpretation if the word is followed by a conjunction and a noun or preceded by a noun and a conjunction (e.g. in "*gauguin moon and earth*", choose the noun interpretation for *moon*, instead of verb[2]).
- In case of ambiguity between adjective and past participle verb, select the adjective interpretation if the word is followed by a noun (e.g. "*stained glass angel*", *stained* is disambiguated as an adjective instead of a past participle verb)

5.3 Chunking

The goal of chunking is to assign a partial structure to a sentence and focuses on easy to parse pieces in order to avoid ambiguity and recursion. In the very specific context of query analysis, and once again since queries have specific linguistic properties (they are not sentences but mostly nominal sequences), chunking can be improved along several heuristics. We propose here some adaptations to improve query chunking to deal with AP and PP attachment, and coordination, using also NE information to guide the chunking strategy.

AP and PP attachment

In standard cases of chunking, AP and PP attachment is not considered, because of attachment ambiguity problems that cannot be solved at this stage of linguistic analysis.

Considering the shortness of queries and the fact that they are mostly nominal, some of these attachments can be solved however in this context.

For the adjectival attachment in French, we attach the post modifier adjectival phrases to the first previous noun with which there is agreement in number and gender. For example, the chunking structure for the query "*Bibliothèque europeenne numerique*" is:

```
NP[ [Bibliothèque AP[europeenne] AP[numerique] ]
```

while it is

```
NP[Bibliothèque] AP[europeenne] AP[numerique]
```

with our standard French grammar.

For PP attachment, we simply consider that the PP attaches systematically to the previous noun. For example, the chunking structure for "*The history of the University of Oxford*" is:

```
NP[the history PP[of the University PP[of Oxford] ] ]
```

instead of:

```
NP[The history] PP[of the University] PP[of Oxford ]
```

Coordination

Some cases of coordination, usually very complex, can be solved in the query context, in particular when NEs are involved. For both English and French, we attach coordinates when they belong to the same entity type (person conj person, date conj date, place conj place, etc.), for example, "*vase achilles et priam*" :

```
NP[vase] NP[Achille et Priam]
```

instead of:

```
NP[vase] NP[Achille] et NP[Priam]
```

[2] *To moon about*

We also attach coordinates when the second is introduced by a reflexive pronoun, such as in: "*[Le laboureur et ses enfants] La Fontaine*" and attach coordinates within a PP when they are introduced by the preposition "*entre*" in French and "*between*" in English.

Use of NE information to guide the chunking strategy

We also use information about NEs present in the queries to guide the query chunking strategy. In standard analysis, NEs are generally part of larger chunking units. In queries, however, because of their strong semantic, they can be isolated as separate chunking units. We have adapted our chunking strategy using this information: when the parser detects a NE (including a date), it chunks it as a separate NP. The following examples show the chunking results for this adapted strategy versus the analysis of standard grammar:

- "*Anglo Saxon 11th century*" (English)

Adapted chunking:
`NP[Anglo Saxon] NP[ 11th century]`

Standard chunking:
`NP[Anglo Saxon 11th century ]`

- "*Alexandre le Grand Persepolis*" (French)

Adapted chunking:
`NP[Alexandre le Grand] NP[Perspolis]`

Standard chunking:
`NP[Alexandre le Grand Perspolis]`

The whole process is illustrated in Figure 1.

When applying the full chain on an example query like "*gauguin moon and earth*", we have the following steps and result:
<u>Preprocessing</u>: *gauguin* is recognized as *Gauguin* (proper noun of celebrity);
<u>Part of speech tagging</u>: *moon* is disambiguated as a noun instead of a verb);
<u>Chunking:</u> *moon and earth* are grouped together in a coordination chunk, *gauguin* is a NE chunked separately.
So we get the following structure:

`NP[Gauguin] NP[moon and earth]`

and *gauguin* is recognized as a person name, instead of

`SC` [3] `[NP[gauguin] FV` [4] `[moon]] and NP[earth],`

gauguin remaining unknown, with the standard English grammar.

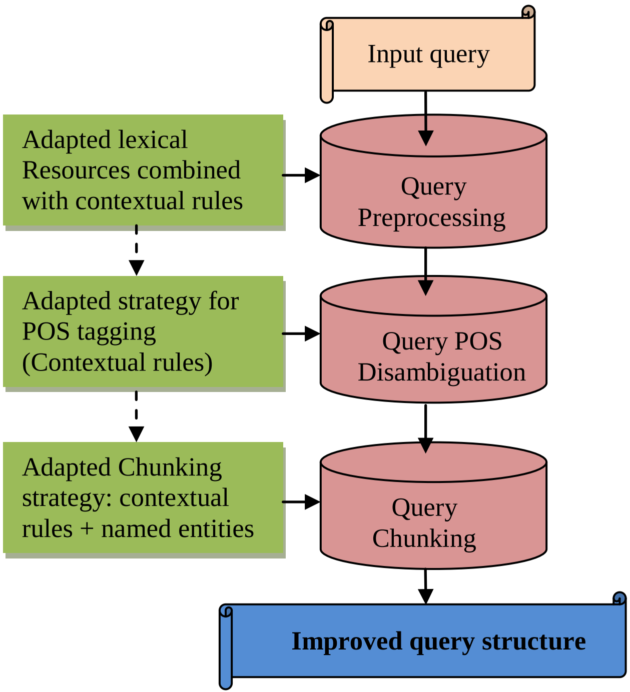
Input query
Adapted lexical Resources combined with contextual rules
Query Preprocessing
Adapted strategy for POS tagging (Contextual rules)
Query POS Disambiguation
Adapted Chunking strategy: contextual rules + named entities
Query Chunking
Improved query structure

Fig 1: Linguistic processing adaptation for queries

5.4 Examples of query structures

The following table shows the differences of query structures obtained with the standard linguistic processing and with the adapted linguistic processing.

1.	Albert Camus la peste
	Standard LP: NP {albert} AP {camus} NP {la peste}
	Adapted LP: NP {albert camus} NP {la peste}
2.	dieux ou héros grec
	Standard LP: NP {dieux} COORD {ou} NP {héros} AP {grec}

[3] SC: chunk tag for sentential clause
[4] FV: finite verb chunk

	Adapted LP: NP {dieux} COORD {ou} NP {héros grec}
3.	pierre bergé
	Standard LP: NP {pierre} VERB {bergé}
	Adapted LP: NP {pierre bergé}

Table 2: Some examples of query structure produced by standard and adapted linguistic processing.

The evaluation of this customization is done indirectly through query translation, and is described in the next section

6 Experiments

6.1 Experimental settings

In our experiments we tried to enrich our baseline SMT system with an adapted linguistic processing in order to improve the query translation. These experiments have double goal. First, to show that the adapted linguistic processing allows to improve query translation compared to a standard linguistic processing, and second, to show that enriching an SMT model with a linguistic processing (adapted) is helpful for the translation.

We use an open source toolkit Moses (trained on Europarl) as a baseline model for query translations. Based on the examples from the section 5, we choose to integrate the chunking and NE information in the translation. We integrate this knowledge in the following way:

- **Chunking:** We check whether the query matches one of the following patterns: “NP1 and NP2”, “NP1 or NP2”, “NP1 NP2”, “NP1, NP2”, etc. If it is the case, the NPs are translated independently. Thus, we make sure that the output query will preserve the logical structure, if “and/or” are treated as logical operators. Also, translating NPs independently might result at different (hopefully better) lexical choices.
- **Named entities:** We introduce XML tags for person names where we propose a possible translation. During the translation process the proposed translation competes with the possible translations from a bi-phrase library. The translation maximizing internal translation score is chosen. In these experiments we propose not to translate an NE at all, however in more general case we could imagine having an adapted NE dictionary.

6.2 Evaluation

We have translated the totality of available Europeana French logs to English (8870 distinct queries), with the following translation models:

- Moses trained on Europarl (Baseline MT)
- Baseline MT model enriched with linguistic processing (as defined in 6.1) based on *basic* grammar (Baseline MT + basic grammar)
- Baseline MT enriched with linguistic processing based on *adapted* grammar (Baseline MT + adapted grammar)

Our approach brings two new aspects compared to simple SMT system. First, an SMT system is enriched with linguistic processing as opposed to system without linguistic processing (baseline system), second: usage of an adapted linguistic processing as opposed to standard linguistic processing. Thus, we evaluate:

1. The impact of linguistic processing on the final query translations;
2. The impact of grammar adaptation (adapted linguistic processing) in the context of query translation.

First, we measure the overall impact of each of the two aspects mentioned above. Table 3 reflects the general impact of linguistic enrichment and grammar adaptation on query structure and translation.

First, we note that the linguistic processing as defined in 6.1 won’t be applied to all queries. Thus, we count an amount of queries out of our test set to which this processing can actually be applied. This corresponds to the first line of the Table 3 (26% of all queries).

Second, we compare the queries translation with and without linguistic processing. This is shown in the second line of the Table 3: the amount of queries for which the linguistic processing lead to different translation (25% of queries for which the linguistic processing was applied).

The second part of the table shows the difference between the standard linguistic processing and an adapted linguistic processing. First, we check how many queries get different structure after grammar adaptation (Section 5) (~42%) and second, we check how many of these queries actually get different translation (~16% queries with new structure obtained after adaptation get different translations).

These numbers show that the linguistic knowledge that we integrated into the SMT framework may impact a limited portion of queries' translations. However, we believe that this is due, to some extent, to the way the linguistic knowledge was integrated in SMT, which explores only a small portion of the actual linguistic information that is available. We carried out these experiments as a proof of concept for the adapted linguistic processing, but we believe that a deeper integration of the linguistic knowledge into the SMT framework will lead to more significant results. For example, integrating such an adapted linguistic processing in a rule-based MT system will be straightforward and beneficial, since the linguistic information is explored directly by a translation model (e.g. in the example 6 in Table 1 tagging "saw" as a noun will definitely lead to a better translation).

Next, we define 2 evaluation tasks, where the goal of each task is to compare 2 translation models. We compare:

1. Baseline MT versus linguistically enriched translation model (Baseline MT+adapted adapted linguistic processing). This task evaluates the impact of linguistic enrichment in the query translation task with SMT.
2. Translation model using *standard linguistic processing* versus translation model using *adapted linguistic processing*. This task evaluates the impact of the adapted linguistic processing in the query translation task.

For each evaluation task we have randomly selected a sample of 200 translations (excluding previously the identical translations for the 2 models compared) and we perform a pairwise evaluation for each evaluation task. Thus, for the first evaluation task, a baseline translation (performed by standard Moses without linguistic processing) is compared to the translation done by Moses + adapted linguistic processing. In the second evaluation task, the translation performed by Moses + standard linguistic processing is compared to the translation performed by Moses + adapted linguistic processing.

The evaluation has been performed by 3 evaluators. However, no overlapping evaluations have been performed to calculate intra-evaluators agreement. We could observe, however, the similar tendency for improvement in each on the evaluated sample (similar to the one shown in the Table 2).

We evaluate the overall translation performance, independently of the task in which the translations are going to be used afterwards (text understanding, text analytics, cross-lingual information retrieval etc.)

Linguistic enrichment	
Nb of *queries* to which the adapted linguistic processing was applied before translation.	2311 (26% of 8870)
Nb of *translations* which differ between baseline Moses and Moses with adapted linguistic processing.	582 (25% of 2311)
Grammar adaptation	
Nb of *queries* which get different *structures* between *standard linguistic processing* and *adapted linguistic processing.*	3756 (42% of 8870)
Nb of *translations* which differ between *Moses+standard linguistic processing* and *Moses+adapted linguistic processing*	638 (16 % of 3756)

Table 3: Impact of linguistic processing and grammar adaptation for query translation

The difference between slight improvements and important improvements as in the examples below has been done during the evaluation.

```
src1: max weber
t1:max mr weber
t2:max weber (slight improvement)

src2: albert camus la peste
t1:albert camus fever
t2:albert camus the plague (impor-
tant improvement)
```

Thus, each pair of translations (t1, t2) receives a score from the scale [-2, 2] which can be:

- 2, if t2 is much better than t1,
- 1, if t2 is better than t1,
- 0, if t2 is equivalent to t1,
- -1, if t1 is better than t2,
- -2, if t1 is much better than t2,

Table 4 presents the results of translation evaluation.

Note, that a part of slight decreases can be corrected by introducing an adapted named entities dictionary to the translation system. For example, for the source query "*romeo et juliette*", keeping NEs untranslated results at the following translation: "*romeo and juliette*", which is considered as a slight decrease in comparison to a baseline translation: "*romeo and juliet*". Creating an adapted NEs dictionary, either by crawling Wikipedia, or other parallel resources, might be helpful for such cases.

Often, the cases of significantly better translations could potentially lead to the better retrieval. For example, a better lexical choice (*don juan moliere* vs. *donation juan moliere*, *the plague* vs. *fever*) often judged as significant improvement may lead to a better retrieval.

Based on this observation one may hope that the adapted linguistic processing can indeed be useful in the query translation task in CLIR context, but also in general query analysis context.

7 Conclusion

Queries posed to digital library search engines in the cultural heritage and social sciences domain tend to be very short, referring mostly to artist names, objects, titles, and dates. As we have illustrated with several examples, taken from the logs of the Europeana portal, standard NLP analysis is not well adapted to treat that domain. In this work we have proposed adapting a complete chain of linguistic processing tools for query processing, instead of using out-of-the-box tools designed to analyze full sentences.

Focusing on the cultural heritage domain, we translated queries from the Europeana portal using a state-of-the-art machine translation system and evaluated translation quality before and after applying the adaptations. The impact of the linguistic adaptations is quite significant, as in 42% of the queries the resulting structure changes. Subsequently, 16% of the query translations are also different. The positive impact of the adapted linguistic processing on the translation quality is evident, as for 99 queries the translation (out of 200 sample evaluated) is improved when compared to having no linguistic processing. We observe also that 78 queries are better translated after adapting the linguistic processing components.

Our results show that customizing the linguistic processing of queries can lead to important improvements in translation (and eventually to multilingual information retrieval and data mining). A lot of the differences are related to the ability of properly identifying and treating domain-specific named entities. We plan to further research this aspect in future works.

	Important ++	**Total nb+**	**Important - -**	**Total nb -**	**Overall impact**
Moses< Moses+ adapted	35	87	4	19	99
Moses+ basic< Moses+ adapted	28	66	2	12	80

Table 4: Translation evaluation. Total nb+ (-): total number of improvements (decreases), not distinguishing whether it is slight or important; important ++ (--): the number of important improvements (decreases). Overall impact = (Total nb+) + (Importan++) – (Total nb-) – (Important --)

Acknowledgements

This research was supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430 (Project GALATEAS).

References

Bin Tan and Fuchun Peng. 2008. Unsupervised query segmentation using generative language models and wikipedia. In Proceedings of the 17th international conference on World Wide Web (WWW '08). ACM, New York, NY, USA, 347-356.

Cory Barr, Rosie Jones, Moira Regelson. 2008. The Linguistic Structure of EnglishWeb-Search Queries, Proceedings of ENMLP'08, pp 1021–1030, Octobre 2008, Honolulu.

James Allan and Hema Raghavan. 2002. Using part-of-speech patterns to reduce query ambiguity. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '02). ACM, New York, NY, USA, 307-314.

Jeann-Pierre Chanod, Pasi Tapanainen. 1995. Tagging French - comparing a statistical and a constraint-based method. Proc. From Texts To Tags:

Issues In Multilingual Language Analysis, EACL SIGDAT workshop. Dublin, 1995.

Jiafeng Guo, Gu Xu, Hang Li, Xueqi Cheng. 2008. A Unified and Discriminative Model for Query Refinement. Proc. SIGIR'08, July 20–24, 2008, Singapore.

Josiane Mothe and Ludovic Tanguy. 2007. Linguistic Analysis of Users' Queries: towards an adaptive Information Retrieval System. International Conference on Signal-Image Technology & Internet–Based Systems, Shangai, China, 2007. http://halshs.archives-ouvertes.fr/halshs-00287776/fr/ [Last accessed March 3, 2011]

Lauri Karttunen. 2000. Applications of Finite-State Transducers in Natural Language Processing. Proceedings of CIAA-2000. Lecture Notes in Computer Science. Springer Verlag.

Marijn Koolen and Jaap Kamps. 2010. Searching cultural heritage data: does structure help expert searchers?. In Adaptivity, Personalization and Fusion of Heterogeneous Information (RIAO '10). Le centre des hautes etudes internationals d'informatique documentaire, Paris, France, 152-155.

Michael Bendersky, W. Bruce Croft and David A. Smith. 2010. Structural Annotation of Search Queries Using Pseudo-Relevance Feedback. Proceedings of CIKM'10, October 26-29, 2010, Toronto, Ontario, Canada

Neil Ireson and Johan Oomen. 2007. Capturing e-Culture: Metadata in MultiMatch., J. In Proc. DELOS-MultiMatch workshop, February 2007, Tirrenia, Italy.

Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. In Proceedings of the 15th international conference on World Wide Web (WWW '06). ACM, New York, NY, USA, 387-396.

Shane Bergsma and Qin Iris Wang. 2007. Learning Noun Phrase Query Segmentation, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 819–826, Prague, June 2007.