Cross Language Retrieval with the Twenty-One system

Wessel Kraaij
TNO-TPD
P.O. Box 155, 2600 AD Delft
The Netherlands
kraaij@tpd.tno.nl

Djoerd Hiemstra
University of Twente/CTIT
P.O. Box 217, 7500 AE, Enschede
The Netherlands
hiemstra@cs.utwente.nl

Abstract

The EU project Twenty-One will support cross language queries in a multilingual document base. A prototype version of the Twenty-One system has been subjected to the Cross Language track tests in order to set baseline performances. The runs were based on query translation using dictionaries and corpus based disambiguation methods.

1 Introduction

1.1 Twenty-One project

Twenty-One is a 2 MECU project with 11 partners [1]funded by the EU Telematics program, sector Information Engineering. The project subtitle is "Development of a Multimedia Information Transaction and Dissemination Tool". Twenty-One started early 1996 and is currently in its building phase.

The Twenty-One database consists of documents in different languages, initially Dutch, English, French and German but extensions to other European languages are envisaged. The TREC Cross Language (CLIR) track task fits our needs to evaluate the system on the aspect of cross language retrieval performance.

1.2 TREC6

Although the development of the full scale Twenty-One system just started in the summer of 1997, Twenty-One accepted the challenge to participate in the cross language track of TREC6.

Whether we would complete the task was a complete question, because at that moment (May 1997), the TNO mono-lingual vector space search engine was still under development and untested, The delivery of a fast workstation was also delayed, and moreover, the consortium was still negotiating with two publishers to acquire bilingual dictionaries. But finally all hard-, soft- and lingware became available just in time to complete some runs in two hectic weeks, without any time for thorough testing.

1.3 Cross Language Retrieval in Twenty-One

The primary approach to Cross Language Retrieval in Twenty-One will be Document Translation (DT). There are certain advantages and disadvantages to DT:

- DT reduces the Cross Language Retrieval task to a monolingual search issue
- The quality of a translation can in principle be better because the full document context is available. In the case of query translation there is often very little context.
- Document translation is slow, but can be done off-line.
- DT requires a full translation of the document base for each supported language, which makes it not really scalable.

The DT approach in Twenty-One will be supplemented with query translation, as a fall-back option and local feedback in the target language for recall enhancement.

A more elaborate description can be found in [2]. However we will test this approach not until TREC7 because the system's partial translation module is not yet finished.

The goal of this year's TREC6 participation (our first participation) is to test the monolingual search

[1] Project partners are: Getronics software, TNO-TPD, DFKI, Rank Xerox Grenoble, University of Twente, University of Tübingen, MOOI foundation, Environ, Climate Alliance, VODO and Friends of the Earth

system and perform baseline runs with dictionary based word translation as a preparation to a full evaluation of Twenty-One within TREC7.

2 Experimental setup

2.1 Retrieval System

The Twenty-One demonstrator[2] system is based on two types of indexes:

- A fuzzy phrase index (n-gram search on phrases extracted from the documents via NLP).
- A standard Vector Space Model (VSM) index based on lemmas

The first index type is well suited for short queries and interactive query refinement, whereas the VSM index is better suited for longer queries. For TREC6 all experiments have been done with the TNO vector space engine. This index employs straightforward *tfidf* weighting and document length normalization. As preprocessing step we used the Xerox morphological tools for tokenization, Part-of-Speech (POS) disambiguation and lemmatization[3]. The dictionary part of the index used for the TREC6 experiments consists of a concatenation of lemma and POS tag. Function words were excluded from the indexing process, based on their POS tag. No traditional stopping list was used.

2.2 Bilingual dictionaries

The translation of the topics was based on a word by word translation process, using the VLIS lexical database from *van Dale* publishers. The VLIS database is a relational database which contains all lexical knowledge that is used for publishing the dictionaries Dutch → foreign language (German, French, English, Spanish). So the database is based on Dutch headwords with translation relations to equivalent lemmas in the foreign languages. The lexical material from the foreign language → Dutch companion dictionaries is not included in the VLIS database. This has some important consequences for its use in a translation system. There are three different types of language pairs:

- Translating from Dutch to a foreign language. This is essentially equivalent to taking the printed version of the van Dale dictionary and looking up each word.
- Translating from a foreign language to Dutch. Although the foreign → Dutch material is not in the database, we can simply lookup Dutch headwords that have the query term as a translation by specifying an appropriate SQL query.
- Translating between two foreign languages. This is simply a combination of the previous types. Look for words in the target language which are a translation of a Dutch lemma which in turn has the query word in the source language as its translation.

The VLIS database contains simple and composite (multi-word) lemmas for 5 languages, Dutch being the pivot language. For Dutch there are 270k entries corresponding to about 513k concepts. These concepts have translations into French, Spanish, German and English.

English	260k	40k	300k
German	224k	24k	248k
French	241k	23k	264k
Spanish	139k	28k	167k

Table 1: Number of translation relations (simple, composite and total) in the Van Dale Lexical database

For TREC6 we only used the simple lemmas. The Xerox morphological tools were used to lemmatize the words in the query in order to find translations.

2.3 Noun phrase corpus

In order to refine the crude word by word translation strategy, a list of Noun Phrases (NP) was compiled from the TREC corpus (the AP88, 89 and 90 data set). The NPs were extracted with the standard NLP tools as used in the Twenty-One system, viz. morphological analysis and POS disambiguation with the Xerox finite state tools followed by NP extraction with the TNO parser. The NPs are not just bigrams but are *maximal*, i.e. they can can contain embedded structures with conjunctions, PP-modification etc. The NPs were sorted and then counted, resulting in a list of unique phrases with frequency of occurrence. As a last step, stopwords were removed.

3 Description of runs

Because the test environment was up and running rather late, we decided to restrict tests to the En-

[2] http://twentyone.tpd.tno.nl/

[3] including compound splitting for German and Dutch

glish document base, but perform cross language experiments with the Dutch, German and French version of the topics. We used no specialized procedure to construct a query from a topic description[4], all runs were fully automatic, full topics (or their translations) were used as queries.

Here's a short description of the runs:

1. A baseline monolingual run: **tnoee**

2. A run based on the MT translated German topics, which were provided by Maryland: **tnodemt**

3. Take the preferred translation from the dictionary: **tno?e1** where ? can be 'd', 'f' or 'nl')

4. Take all translations from the dictionary, i.e. each topic word is substituted by a list of all translations from the dictionary: **tno?e2**

5. Mark the Noun Phrases in the original topic. Subsequently replace each word by a list of its translations. This results in a multitude of possible translations of each NP. The possible translations are disambiguated using the NP corpus which was described in the previous subsection. Section 4 describes the disambiguation procedure in more detail. Finally queries are constructed, either by:

    - mapping translation probabilities into term weights: **tno?e4**
    - taking the most probable translation: **tno?e3**

4 Disambiguation

Disambiguation of the translated NPs is based on candidate NPs extracted from the document base. The introduction of NPs (or any multi-word expression) in the translation process leads to two types of ambiguity: sense ambiguity and structural ambiguity (or underspecification) which are displayed in a data structure called a translation chart.

Figure 1 gives the French translation chart of the English NP *third world war*. Each word in this NP can have several translations that are displayed in the bottom cells of the chart, the so-called sense ambiguity. According to a list of French NPs there may be two candidate multi-word translations: *tiers monde* for the English NP *third world* and *guerre mondiale* for *world war*. These candidate translations are displayed in the upper cells of the chart. Because the internal structure of NPs was not available for the translation process, we can translate a full NP by decomposing it in several ways. For example *third world war* can be split up in the separate translation of either *third world* and *war* or in the separate translation of *third* and *world war*.

-		
tiers monde	guerre mondiale	
troisième tiers	monde mondiale terre	guerre bataille
third	world	war

Figure 1: translation chart of *third world war*

The chart of figure 1 represents a total of 12 possible translations of which only one is *troisième guerre mondiale*. Constructing the translation chart and finding the most probable translation was done as follows.

1. The query is tagged and NPs are extracted from it. The disambiguation procedure is only used to disambiguate the NPs from the query

2. During dictionary look-up the bottom cells of the translation chart are filled. (Later on in the project, dictionary look-up can be extended with the composite lemmas from the dictionary.)

3. The upper cells of the translation chart are filled with candidate NPs that contain words of the corresponding bottom cells. If possible translations of two (or more) cells coocurred in an extracted NP, the possible translations are treated as a candidate NP.

4. Probabilities are assigned to the candidate NPs in each cell of the translation chart. Probabilities are based on the frequency of the candidate NP in the document base and on the contents of the dictionary. In the final version of the Twenty-One system, information from parallel corpora will also be used to estimate probabilities [1].

5. Take the most probable candidate NP that contains possible translations of each word of the query NP.

[4]Query stopwords like *document* and *relevant* were not excluded

6. If there is no such candidate NP repeat step 5 for $n = 2$ candidate NPs. If there is still no match back-off to $n + 1$ NPs until a match is found.

For the example of figure 1 the algorithm has to back-off once because there is no candidate NP that covers the translation of all the words of the query NP (the top of the chart is empty). After one back-off step there is still some ambiguity left. Queries can be constructed either by mapping the probabilities of the translations into term weights or by taking the most probable translation.

5 Discussion

5.1 Results

run name	average prec.	performance relative to baseline (%)
tnoee	0.2752	100
tnode1	0.1453	53
tnode1-fix	0.1721	62
tnode2	0.0568	20
tnode2-fix	0.0977	35
tnode3	0.2090	76
tnode4	0.2013	73
tnodemt1	0.0977	35
tnofe1	0.0913	33
tnofe1-fix	0.1131	41
tnofe2	0.0477	17
tnofe2-fix	0.0498	18
tnofe3	0.1403	51
tnofe4	0.1305	47
tnonle1	0.0841	30
tnonle1-fix	0.1545	56
tnonle2	0.0733	26
tnonle2-fix	0.0972	35
tnonle3	0.1930	70
tnonle4	0.1729	62

Table 2: Results

Table 2 lists the the non interpolated average precision and the relative performance with respect to the baseline version **tnoee**.[5]

5.2 Preprocessing bugs

The results gave us reason to have another look at the translated queries for the different languages. Due to the enormous time constraints our system still contained some minor bugs that affected the CL results of all three languages, e.g. wrong handling of capital letters, hyphens, diacritial markers, etc. One of these minor bugs had major implications: the character $ (used as an escape character in one of the intermediate formats) caused a lot of not relevant hits, because it was not removed in all the runs.

In the table we included unofficial bugfix runs for the runs labelled '1' and '2'. These runs (in particular tnode2, tnofe1, tnofe2, tnonle1, tnonle2 and also the runs '5' and '6' which are not listed in the table) all suffered severely from the '$-bug'.

The lexical lookup and tokenizing process is still far from perfect though. Especially the handling of compounds, geographical names and diacritics needs to be improved for TREC7.

5.3 Fundamental problems

A first look at the translated queries also gives some indication of errors that are not due to bugs in our implementation, but due to our approach to CLIR.

multi-word expressions Not using the multi-word expressions from the van Dale lexical database is probably the most important source of errors. It leads to obvious errors like the wrong translation of e.g. *pommes de terres*. It also leads to errors in the translation of phrases that seem to exist of word by word translations, like e.g. *deuxième guerre mondiale* which is in English *second world war*. In French *mondiale* is an adjective and possible translations are *worldwide* and *global* but not the noun *world*. Of course, if the correct translation is not among the possible translations the disambiguation procedure will not find it either. (the multi-word expression *world war* does have an entry in van Dale.)

Proper names Because we did not use a module for proper name recognition, the system will try to translate them, which for instance leads to the translation of *Kurt Waldheim* into *Kurt forest home*.

Tagger errors The current system performs syntactic disambiguation before dictionary look-up (the Xerox tagger) and sense diambiguation after dictionary look-up. The Xerox tagger will make a small percentage of errors during the tagging process which leads to wrong translations. Maybe skipping syntactic disambiguation would be beneficial, because there is a final disambiguation step in the target language.

[5]The average precision has been computed on the basis of only 22 of the 25 topics

5.4 MT vs. dictionary look-up

The LOGOS MT run does underperform suprisingly. Upon closer inspection we found that a lot of its bad performance can be attributed to lack of robustness with respect to tokenization, compound handling, and most importantly by gaps in its dictionary. Common but vital topic terms like 'Parfum', 'Baumwolle' en 'Akupunktur' were left untranslated.

6 Conclusion & Outlook

We have succeeded in building a CLIR system which performs above median for most runs. We believe the performance of the monolingual system can be siginificantly improved by incorporating the latest weighting methods, tuning stoplist and some more attention to topic preprocessing.

The general picture of our CLIR runs is that taking the preferred translation from the dictionary works better than taking all translations with equal probability. But more important, the corpus based disambiguation technique seems to result in significant improvements. We don't know yet how much of this improvement is due to the phrase context. It's also not clear whether taking the most probable translation is better than taking the probability vector as the translation for each term.

Although it's easy to produce a table filled with average precision figures, it's hard to draw conclusions about the relative merits of the different systems and methods. The quality of a significant part of the topic translations provided by NIST and CLIR participants is not without errors or omissions, which makes comparisons across languages less meaningful (even comparing to the English baseline). The variance of the results among the topics is also extremely high because of gaps in the translation dictionaries. This makes a comparison of CLIR methodologies based on different dictionaries[6] an impossible task. Supplying a base-line dictionary (like the base-line Speech Recognizer results delivered by NIST in the SDR track) would enable a more meaningful comparison of dictionary based methods. Otherwise CLIR participants might find themselves comparing the coverage of their dictionaries instead of comparing methods for CLIR.

Acknowledgements

We would like to thank all colleagues working on Twenty-One . In particular we want to thank: Rudie Ekkelenkamp, Jurgen den Hartog for their work on the search engine (both TNO), Hervé Poirrier, Anne Schiller and David Hull of RXRC for help with the Xerox morphological tools and general advice, Franciska de Jong and UT students for translating the queries to Dutch, Tillman Wegst of DFKI for the integration of the Xerox morphological tools with the TNO parser.

[6] e.g. between our van Dale runs, the LOGOS MT run and runs from other groups

References

[1] Djoerd Hiemstra, Franciska de Jong, and Wessel Kraaij. A domain specific lexicon acquisition tool for cross-language information retrieval. In L. Devroye and C. Chrisment, editors, *Proceedings of RIAO'97*, pages 217–232, 1997.

[2] Wessel Kraaij. Multilingual functionality in the TwentyOne project. In David Hull and Douglas Oard, editors, *AAAI Symposium on Cross-Language Text and Speech Retrieval*. American Association for Artificial Intelligence, March 1997. http://www.clis.umd.edu/dlrg/filter/sss/papers/.

7 Questionnaire

```
1. OVERALL APPROACH:
----------------------

  1.1 What basic approach do you take to cross-language retrieval?
      [X] Query Translation  IN TREC6
      [X] Document Translation : in the project and probably in TREC7
      [ ] Other, _______________________


  1.2 Were manual translations of the original NIST topics used as a
      starting point for any of your cross-language runs?
      [X] No
      [ ] Yes, _______________________

  1.3 Were the automatically translated (Logos MT) documents used
      for any of your cross-language runs?
      [X] No
      [ ] Yes, _______________________

  1.4 Were the automatically translated (Logos MT) topics used
      for any of your cross-language runs?
      [ ] No
      [X] Yes, run tnodemt1



2. MANUAL QUERY FORMULATION:
------------------------------

No manual query formulation.



3. USE OF MANUALLY GENERATED DATA RESOURCES:
---------------------------------------------

  3.1 What kind of manually generated data resources were used?
      [X] Dictionaries
      [ ] Thesauri
      [X] Part-of-speech Lists
      [X] Other, Lemmatizers

  3.2 Were they generated with information retrieval in mind or were
      they taken from related fields?
      [ ] Information Retrieval
      [ ] Machine Translation
      [X] Linguistic Research
      [X] General Purpose Dictionaries
      [ ] Other, _______________________

  3.3 Were they specifically tuned for the data being searched (ie.
```

```
    with special terminology) or general-purpose?
    [ ] Tuned for data; Please specify _______________________
    [X] General purpose

3.4 What amount of work was involved in adapting them for use in
    your information retrieval system.

    Dictionaries: 3 days
    Morphology: 3 days

3.5 Size
   For dictionary size cf. table 1.in the paper.

3.6 Availability? - Please also provide sources/references!
    [X] Commercial: Xerox Xelda toolkit
    [X] Proprietary: Van Dale dictionaries
    [ ] Free
    [ ] Other, _______________________



4. USE OF AUTOMATICALLY GENERATED DATA RESOURCES:
---------------------------------------------------

  4.1 Form of the automatically constructed data resources?
      [ ] Lexicon
      [ ] Thesaurus
      [ ] Similarity matrix
      [X] Other, List of Noun Phrases extracted from the corpus

  4.2 What sort of training data was used to construct them?
      [X] Same data as used for searches, _______________________
      [ ] Similar data as used for searches, _______________________
      [ ] Other data, _______________________

  4.3 Size
      [ ] 4.4 million ___________________ entries
      [ ] 128 MBytes

  4.4 Was there any manual clean-up involved in the construction process?
      [ ] Yes, _______________________
      [X] No

  4.5 Rough resource estimates for building the data resources (ie. an
      indicator of the computational complexity of the process).
      [10] (Sparc Ultra 300 Mhz) hours
      [ ] _______________________ MBytes of memory used
      [ ] _______________________ temporary disk space



5. GENERAL
------------
```

5.1 How dependent is the system on the data resources used? Could they easily be replaced if better sources were available?
[ ] Very dependent, _______________________
[ ] Somewhat dependent, _______________________
[X] Easily replacable, _______________________
[ ] Don't know

5.2 Would the approach used potentially benefit if there were better data resources (e.g. bigger dictionary or more/better aligned texts for training) available for tests?
[ ] Yes, a lot, _______________________
[X] Yes, somewhat, _______________________
[ ] No, not significantly, _______________________
[ ] Don't know

5.3 Would the approach used potentially suffer a lot if similar data resources of lesser quality (noisier dictionary, wrong domain of terminology) were used as a replacement?
[ ] Yes a lot, _______________________
[X] Yes, somewhat, _______________________
[ ] No, not significantly, _______________________
[ ] Don't know

5.4 Are similar resources available for other languages than those used?
[X] Yes, Spanish
[ ] No