An Efficient Patent Keyword Extractor As Translation Resource

Svetlana Sheremetyeva
LanA Consulting ApS
Møllekrog, 4, Vejby
3210 Copenhagen, Denmark
lanaconsult@mail.dk

Abstract

The paper addresses the issue of resource reuse in patent translation. It presents an efficient patent keyword/phrase extraction tool and illustrates how the tool can be used in patent translation by both human experts and MT developers. The keyword extraction is based on a new hybrid methodology providing for intelligent output and computationally attractive properties. The tool is composed of two modules, - an NP extractor and a patent-tuned scoring module. It does not require a corpus for calculating keywords and relies only on statistical information in a single document which is an advantage for developers and users who would not depend on the corpus availability. The approach is portable across domains, languages and applications.

1 Introduction

The quality of both human and machine translation in the patent domain where new terminology emerge every day is to a large extent influenced by the quality and operative maintenance of lexical databases. This is exactly the area where automatic keyword extraction and especially noun phrase extraction can be used to a great advantage.

NPs can often be translated irrespective of the context which significantly reduces the time and effort of human translators and MT developers. However, despite a lot of research on automatic extraction the problem still presents a tough challenge (Piao et al., 2005). Various data and text mining tools applied to patent analysis and relying on keyword extraction procedures have been around for quite a while now (Duda et al., 1973; Hehenberger et al., 1998; Matsuo et al., 2003; Fattori et al., 2003; Tseng Yuen-Hsien et al., 2005). Following, in general, the hybrid technique trends in the field they try to get more relevant results by using domain restrictions, such as the strict structuring of a patent and its linguistic constraints.

In our paper we present KeyPat, - a tool for extracting noun keywords/phrases from patents and illustrate how the tool can be used in patent translation by both human experts and MT developers.

KeyPat is composed of two modules, - an NP extractor based on methodology described in (Sheremetyeva, 2009), and a patent-tuned scoring module. KeyPat does not require a corpus for calculating keywords and relies only on statistical information in a single document. This is an advantage for users who would not depend on the corpus availability (Matsuo et al., 2003). The tool can be used both by software developers and directly by human translators and patent experts.

The paper is structured as follows: Section 2 gives a task definition; Section 3 includes an overview of the parent NP extractor; Section 4 presents the KeyPat algorithm and scoring module; Section 5 describes the KeyPat interactive user interface. Section 6 discusses possible ways of using KeyPat in translation and describes a cross-language portability experiment. Finally, some conclusions are given.

2 Task definition

The target of our extraction effort is defined by the intersection of the following criteria: (i) grammaticality, (ii) noun, (iii) multiword expression, (iv) terminology, (v) relevancy scoring, (vi) processing speed, (vii) robustness and user friendliness.

Our intention is to extract keywords that can be useful in both machine and human patent processing. Within professional patent business it is always a patent analyst who makes the final judgment on the relevancy of a patent under examination or on the quality of its translation. We will thus try to extract keywords with "a human face" as well-formed grammatical units.

Our ambition is to extract noun phrases. We recognized them through the usual criteria regarding their morphosyntactic context. We consider a word string be a multiword expression if the possibility of computing their meaning from their elements is a worse solution than storing them in lexicons (Laporte et al., 2008). We assume that multiword noun expressions in a technical text are terms (Daille, et al., 2008). To make extracted NP terms more relevant we do not aim to extract an NP, - part of a longer NP if the shorter phrase does not function individually in the processed corpus or text. The two large narrative text fields in a patent dominate the distribution of important keywords, - DESCRIPTION presenting an invention in detail and CLAIMS, the most important part of a patent which has legal meaning. CLAIMS texts often start with PREAMBLEs containing information on the prior art of an invention. Patent analysts consider terms used in CLAMS and especially in its PREAMBLEs most closely content-related and, hence, relevant. We kept this in mind when developing a scoring algorithm.

Patent corpora are changing every day as more and more patents are being constantly filed. As it is hardly possible to get ever renewing patent corpora operatively enough we will rely on statistical and linguistic information in a single document assuming corpus unavailable.

3 NP extraction

We reuse the hybrid NP extraction methodology described in (Sheremetyeva, 2009) that provides for grammaticality, robustness and is attractive calculation-wise.

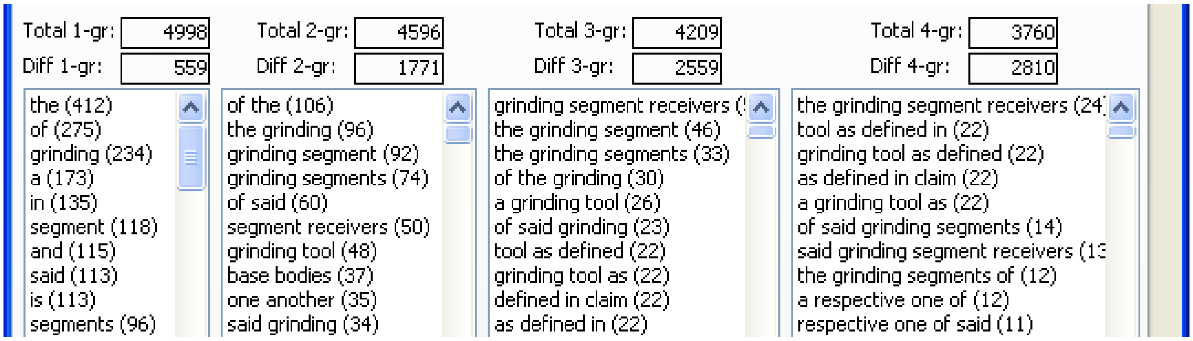
NP extraction is based on statistical techniques merged into a strongly lexicalized Constraint Grammar paradigm with a corpus-based domain lexicon at its core.

The key ideas here are as follows:

- Calculation of n-grams (0<n<5)[1] as the initial set of NP candidates is done on a raw text without stoplists and lemmatization as preprocessing. Starting extraction with stoplist words removal may lead to "bad" combinations of words. For example, the removal of stoplist words (boldfaced) at the preprocessing stage from the patent fragment: "**a** table **in which the** wireless location system… " will lead to the extraction of such stings as "*table wireless location system" which are not terms in the current document and can be misleading. Heuristic stemming algorithms, may fail to identify inflectional variants and lead to the extraction of wrongly combined and/or truncated character strings making them unreadable for a human expert. Proper NLP lemmatization with more correct results might have coverage problems and is very expensive computationally. To bypass these problems lemmatization is postponed to the very last stage of extraction. It has the advantage of making it possible to reduce lemmatization to nouns only.
- Filtering the initial set of candidates by discarding those n-grams which cannot be NPs, rather than searching n-grams matching NP patterns which is achieved by
    - breaking the lexicon into parts, each containing wordforms belonging to certain parts-of-speech which cannot be found at the starting, middle and end positions in the NP pattern;
    - building a special constraint grammar whose rules are only applied to n-gram components rather than to whole n-grams;
    - applying the grammar rules to n-gram components through direct lexical (word string) match against a corresponding lexicon part rather than through POS tagging.

[1] This is the most widely used limit for the number of words in n-gram extraction, but in our system "n" can be set to any number.

(i)



Total 1-gr: 4998
Diff 1-gr: 559

the (412)
of (275)
grinding (234)
a (173)
in (135)
segment (118)
and (115)
said (113)
is (113)
segments (96)

Total 2-gr: 4596
Diff 2-gr: 1771

of the (106)
the grinding (96)
grinding segment (92)
grinding segments (74)
of said (60)
segment receivers (50)
grinding tool (48)
base bodies (37)
one another (35)
said grinding (34)

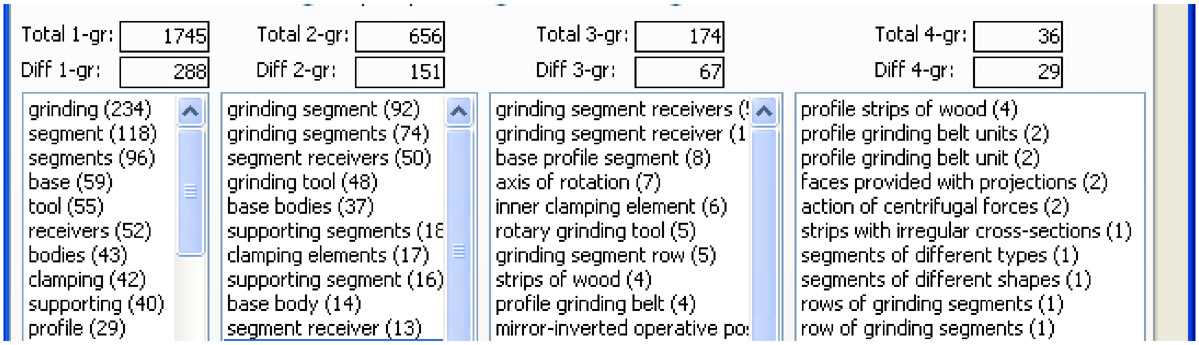
Total 3-gr: 4209
Diff 3-gr: 2559

grinding segment receivers (
the grinding segment (46)
the grinding segments (33)
of the grinding (30)
a grinding tool (26)
of said grinding (23)
tool as defined (22)
grinding tool as (22)
defined in claim (22)
as defined in (22)

Total 4-gr: 3760
Diff 4-gr: 2810

the grinding segment receivers (24
tool as defined in (22)
grinding tool as defined (22)
as defined in claim (22)
a grinding tool as (22)
of said grinding segments (14)
said grinding segment receivers (13
the grinding segments of (12)
a respective one of (12)
respective one of said (11)

(ii)



Total 1-gr: 1745
Diff 1-gr: 288

grinding (234)
segment (118)
segments (96)
base (59)
tool (55)
receivers (52)
bodies (43)
clamping (42)
supporting (40)
profile (29)

Total 2-gr: 656
Diff 2-gr: 151

grinding segment (92)
grinding segments (74)
segment receivers (50)
grinding tool (48)
base bodies (37)
supporting segments (18
clamping elements (17)
supporting segment (16)
base body (14)
segment receiver (13)

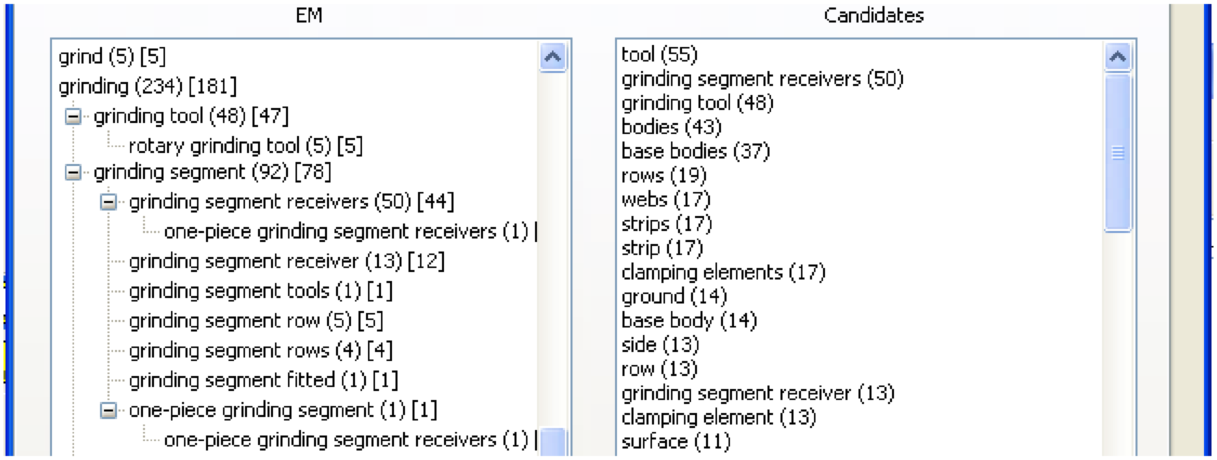
Total 3-gr: 174
Diff 3-gr: 67

grinding segment receivers (
grinding segment receiver (1
base profile segment (8)
axis of rotation (7)
inner clamping element (6)
rotary grinding tool (5)
grinding segment row (5)
strips of wood (4)
profile grinding belt (4)
mirror-inverted operative po

Total 4-gr: 36
Diff 4-gr: 29

profile strips of wood (4)
profile grinding belt units (2)
profile grinding belt unit (2)
faces provided with projections (2)
action of centrifugal forces (2)
strips with irregular cross-sections (1)
segments of different types (1)
segments of different shapes (1)
rows of grinding segments (1)
row of grinding segments (1)

Figure 1. Fragments of the top 1- to 4-gram lists (i) calculated over a raw patent text (top screenshot) and (ii) after filtering with the grammar rules (bottom screenshot). Numbers in brackets show frequencies.



EM

- grind (5) [5]
- grinding (234) [181]
  - ⊟ grinding tool (48) [47]
    - rotary grinding tool (5) [5]
  - ⊟ grinding segment (92) [78]
    - ⊟ grinding segment receivers (50) [44]
      - one-piece grinding segment receivers (1) [
    - grinding segment receiver (13) [12]
    - grinding segment tools (1) [1]
    - grinding segment row (5) [5]
    - grinding segment rows (4) [4]
    - grinding segment fitted (1) [1]
    - ⊟ one-piece grinding segment (1) [1]
      - one-piece grinding segment receivers (1) [

Candidates

tool (55)
grinding segment receivers (50)
grinding tool (48)
bodies (43)
base bodies (37)
rows (19)
webs (17)
strips (17)
strip (17)
clamping elements (17)
ground (14)
base body (14)
side (13)
row (13)
grinding segment receiver (13)
clamping element (13)
surface (11)

Figure 2. A fragment of the expansion matrix of the 1-gram "grinding (left) and the unlemmatized list of keyword candidates filtered with the U criterion (right). Shown in round brackets are frequencies, in square brackets – the number of sentences in which an n-gram occurred.

4 KeyPat

4.1 Algorithm

The KeyPat top algorithm of keyword/phrase extraction is as follows:

1. PATENT SEGMENTATION
2. IDENTIFICATION OF CANDIDATES
    a. Calculating raw text n-grams (n=1,2,3,4)
3. NP FILTERING
    a. Filtering out candidates with the use of the lexicalized grammar rules
    b. Filtering out shorter candidates which are parts of longer candidates and are not used individually in a given patent
4. RELEVANVCY FILTERING
5. LEMMATIZATION
6. OUTPUT.

The keywords are calculated separately for the DESCRIPTION and CLAIMS parts of a patent and then merged. Therefore we first segment a patent into 2 large parts, - DESCRIPTION and CLAIMS which is pretty straightforward due to the patent structure standard. The CLAIMS part is then segmented into separate sentences which can also be done with high reliability as every claim in the CLAMS part is necessarily structured as one sentence. The beginning (and end) of a claim text can be identified due to the patent formatting. Segmenting PREAMBLEs in CLAIMS texts is not that easy and we developed a special procedure that involves certain linguistic knowledge. As for the DESCRIPTION part it contains a lot of special symbols involving comas, periods, capitalization, etc., which are ambiguous with sentence borders and thus make sentence splitting extremely unreliable. We therefore do not split the DESCRIPTION part into sentences and use co-occurrence information imbedded in other calculation parameters.

The quality of NP filtering with the lexicalized grammar rules can be judged by comparing the n-gram lists in Figure 1. We use the US patent # 04777771 for illustration.

To make a decision whether shorter NP candidates included in longer NPs function on their own in the processed patent and thus "have the right" to be included in the output an expansion matrix is built (Figure 3; right).

The matrix is created for every top 1-gram[2] word by nesting string-overlapping phrases which are then filtered by a count-based criterion "Uniqueness" (U).

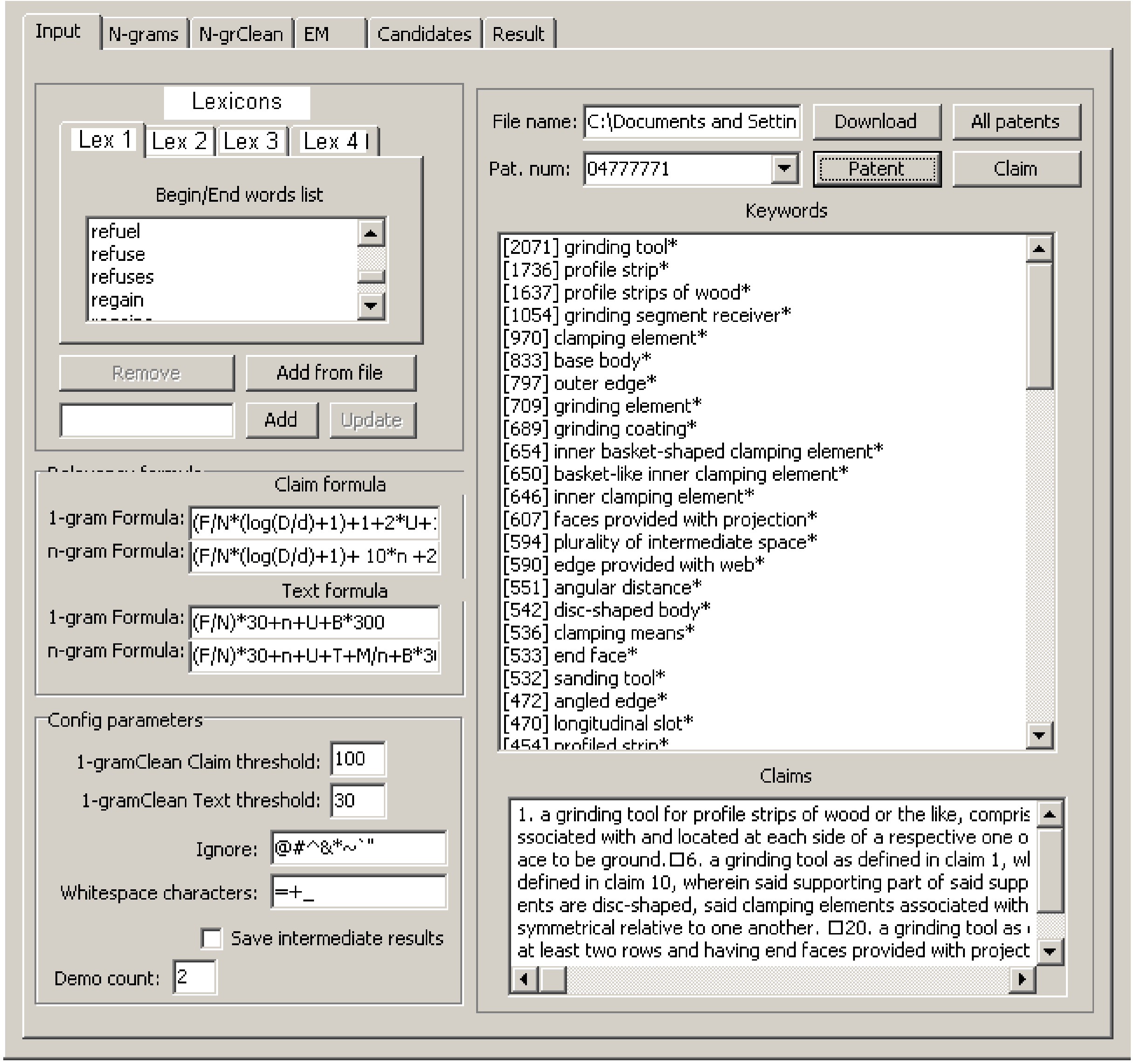
"Uniqueness" is defined as the difference between an n-gram frequency and the sum of frequencies of its (n+1)-gram expansions. A low U-value shows that the candidate is unlikely to be used individually. The U=0 or U< 0 values are used as thresholds for filtering out undesired candidates. The tool can be switched to a rougher mode of filtering, - just delete candidates which are parts of longer ones. Then one can save computation time by skipping the calculation of extension matrix and U-criterion.

4.2 Scoring Module

In scoring we take into consideration the location of a keyword (DESCRIPTION, CLAIMS, and PREAMBLE) and its statistical information. Relevancy vector values are calculated as a combination of mathematical operations (+, -, x, / and log) over the scoring parameters and any integer coefficients. The full range of the KeyPat scoring parameters is given below:

F – keyword frequency;
N - average frequency of n-grams;
n - n-gram length;
U - uniqueness; the difference between an n-gram frequency and the sum of frequencies of its (n+1) - gram expansions;
T - n-gram composition rate; shows how many top list 1-grams a given n-gram contains;
M - n-gram matrix index; the sum of frequencies of words included in an n-gram;
D - number of sentences in the text;
d - number of sentences where a given n-gram occurred at least once;
C - belongs to CLAIMS; shows that an n-gram occurs in patent CLAIMS at least once;
B - belongs to PREAMBLE shows that an n-gram occurs in a CLAIMS text PREAMBLEs at least once.

[2] Depending upon the user settings the extension matrix can be calculated for any number of top (or all) 1-grams



refuel
refuse
refuses
regain

Claim formula

1-gram Formula: (F/N*(log(D/d)+1)+1+2*U+:

n-gram Formula: (F/N*(log(D/d)+1)+ 10*n +2

Text formula

1-gram Formula: (F/N)*30+n+U+B*300

n-gram Formula: (F/N)*30+n+U+T+M/n+B*3

Config parameters

1-gramClean Claim threshold: 100

1-gramClean Text threshold: 30

Ignore: @#^&*~`"

Whitespace characters: =+_

☐ Save intermediate results

Demo count: 2

File name: C:\Documents and Settin

Pat. num: 04777771

```
[2071] grinding tool*
[1736] profile strip*
[1637] profile strips of wood*
[1054] grinding segment receiver*
[970] clamping element*
[833] base body*
[797] outer edge*
[709] grinding element*
[689] grinding coating*
[654] inner basket-shaped clamping element*
[650] basket-like inner clamping element*
[646] inner clamping element*
[607] faces provided with projection*
[594] plurality of intermediate space*
[590] edge provided with web*
[551] angular distance*
[542] disc-shaped body*
[536] clamping means*
[533] end face*
[532] sanding tool*
[472] angled edge*
[470] longitudinal slot*
[454] profiled strip*
```

```
1. a grinding tool for profile strips of wood or the like, compris
ssociated with and located at each side of a respective one o
ace to be ground.□6. a grinding tool as defined in claim 1, wl
defined in claim 10, wherein said supporting part of said supp
ents are disc-shaped, said clamping elements associated with
symmetrical relative to one another. □20. a grinding tool as
at least two rows and having end faces provided with project
```

Figure 3. A screenshot of the KeyPat user interface with the keywords extracted for the US patent # 04777771. Numbers in square brackets are relevancy scores. The asterisk indicates that the term occurred both in the CLAIMS and DESCRIPTION parts of the patent. Shown in the left pane are tool settings which can be changed by the user.

The parameters “M” (n-gram matrix index) and “T” (n-gram composition rate) are not relevant for 1-gram candidates, as for a 1-gram “M” coincides with “F” and “T” is always 1. The parameters “D” and “d” (number of sentences) are only relevant for calculating keywords over the CLAIMS. As was explained in Section 4.2 DESCRIPTION is not spitted into sentences.

The DESCRIPTION and CLAIMS keywords are calculated and scored separately and then merged. Due to the different applicability of parameters to 1-grams and 2-4-grams and partwise we use 4 different empirically determined scoring vectors for 1-gram as opposed to 2-4-grams depending on their location.

For example, the 2-4-gram relevancy vector (R) of a keyword phrase found in CLAIMS is calculated as follows:

$$R=(F/N(\log(D/d)+1)+10n+2U+M/n+ +10C+500B$$

Relevancy filtering is performed based on a certain threshold set for the R value.

The final KeyPat output for our example sorted according to its relevancy is shown in Figure 3. The "*" sign means that the n-gram was found in CLAIMS part of the patent which, in a correctly composed patent text, means that the keyword also appeared in DESCRIPTION.

As a by-result this "*" feature can be used for checking the terminology consistency and correctness of patent text composition. According to Patent Law CLAIMS should not include terms not mentioned in the DECSRIPTION part. KeyPat has a functionality to make this check automatically.

5 KeyPat User Interface

KeyPat is programmed in C++ and has two implementations: (i) for the Windows and (ii) Linux operational environments.

A screenshot of the KeyPat user interface (available in the Windows implementation) is shown in Figure 3. The buttons above the keyword window make it possible to process (i) a single patent text as a whole, (ii) its CLAIMS only (which some of the human experts prefer) and, (iii) any number of patens or their claims loaded from a user patent database, keywords for every patent being saved automatically. The right pane bottom window displays the CLAIMS text of a processed patent for the user control.

KeyPat is implemented as a tool with flexible settings which allows tuning the tool to specific domains and user preferences.

The left pane of the interface contains interactive text areas for changing the KeyPat knowledge. The lexicon bookmarks allow the user to change the basic KeyPat lexicons to better suit the domain or purpose of extraction. Any word added to one or several of these lexicons will lead to the removal of n-grams containing this word in a certain position according to the KeyPat grammar. For example, if the user does not want to have such generic single words as "system", "apparatus", etc., in the final output (which will most probably happen due to the high frequency of such words) it is enough to add them to lexicon 4 which takes care of undesired or "bad" phrases which passed the filtering procedures. Actually, if desired, changing the content of the lexicons allows for tuning KeyPat to extracting the part-of-speech types of phrases other than NPs.

The "formula" text areas in the middle of the left pane let the user set his/her scoring vectors within the range of given parameters and operations over them (see Section 4.2). The "threshold" areas allow for the keyword calculation over different top sets of 1-gram frequency lists resulting from grammar filtering. In Figure 3 KeyPat is set to include all 100% of "clean" 1-grams from the CLAIMS part and 30% of the top frequency "clean" 1-grams from the DESCRIPTION part.

The bookmarks on the top of the interface open pages with the intermediate results of calculations as shown in Figures 1 and 2 which can be used for the user control and/or research in text mining.

6 KeyPat in Patent Text Translation and Portability Experiment

The most evident and direct use of KeyPat in translation is knowledge acquisition.

For example, its current product-level English version can be used by both human translators and MT developers.

When translating into English which is a necessary step in international patenting for any national application, a human translator who is normally not a patent expert can operatively get the most contemporary terminology by running KeyPat over English patents in the corresponding domain. The author can refer to the author's own successful experience in using KeyPat for this purpose.

As for MT development KeyPat can be used for text mining in creating unilingual (English in this case) lexicons as the first and basic stage in creating multilingual MT resources (Daille and Morin, 2008). In many cases, independent of the translation direction, especially for low resource languages, it is the English side that lexical acquisition starts from. For example, (Hewavitharana et al, 2007) developing an Arabic-to-English MT system extract English NPs at the starting point of their development as the available Arabic parsers do not produce desired accuracy. In this respect both RBMT systems and phrase-based SMT systems can benefit from the KeyPat techniques and its direct output.

One more possible area of using the current version of KeyPat in MT development is to apply it to the analysis stage in systems with English-to-non-English translation directions. One can think of running KeyPat over a source text at the preprocessing stage and then applying its output to the source text analysis (NP chunking), but this, of course, might be problematic and needs further research.

In the multilingual perspective it is possible to extent KeyPat to other languages.

In spite of the fact that the essential part of the tool knowledge contains language-dependent information, - lexicons and linguistic knowledge on the language NP structure, the specificity of using this knowledge makes KeyPat extraction algorithm in general language independent. All extraction procedures, - n-gram calculation, removal of n-grams whose individual components match KeyPat lexicon, construction of extension matrix, filtering with the U criterion and finally relevancy calculation are fully portable across languages. The difference in linguistic knowledge about noun phrases in different languages (normally available) can be completely captured by the slight adaptation of the constraint grammar rules based on the knowledge about the NP word order and content of KeyPat lexicons, which of course still need to be acquired. However, it should not be difficult as these lexicons are shallow, - just the lists of wordfoms sorted into different part-of-speech classes, for which available automated tools can be used.

We carried out a feasibility experiment in porting the English KeyPat to the French language. To account for the differences in the English and French NP structures it was mainly necessary to only take care of the different Adverb-Adjective-Noun order in English and French as, e.g., in

"rigorously_Adv constant_Adj speeds_N" → "vitesses_N rigoureusement_Adv constants_Adj"

We reused the English KeyPat lexicon part-of-speech classification inventory for French with the exception of the use of adjectives and adverbs. Into French lexicon 1 that should contain wordfoms which are forbidden at the beginning of the French noun phrase adjectives and adverbs were included, as according to the French grammar adjectives and adverbs should not normally (at least in terminology) be put in the first position in NP. On the contrary, French lexicon 2 "responsible" for deleting n-grams with "wrong" words in the middle position unlike English does not include adjectives and adverbs as in the French NP they can occur in the middle position. For the same reason adjectives are not included in the brush-up lexicons 3 cleaning phrases ending in any part of speech but nouns, - a French NP can end in an adjective. The part-of-speech inventory of lexicon 4 which is used to remove single words which are not nouns was ported without change. The fact that nouns are not required in any of the KeyPat lexicons (nouns can occur in any NP position) greatly reduced the lexicon acquisition effort and made French KeyPat even with a limited training knowledge quite robust. To acquire separate part-of-speech lists form a French training corpus we used built-in-house tools based on part-of-speech morphology followed by human refinement. The results of the experiment were quite good and confirmed a KeyPat portability potential.

It is natural to further investigate whether KeyPat can be used for automatic multilingual lexicon acquisition. We ran the tool on parallel French – English patent descriptions applying the same scoring vectors and got English and French keywords lists mainly equivalent but not aligned. The lack of alignment is due to different word counts in the equivalent English and French NPs because of language discrepancies and insufficient tool training. One possible way to overcome these problems is to apply alignment techniques to parallel bilingual/multilingual keyword lists. In this respect further experiments are needed.

7 Conclusions

In this paper, we described an efficient Patent NP keyword extractor based on a hybrid methodology that can be used by human translators, patent experts and MT developers.

The keyword extraction procedure does not require corpus or elaborated grammar rules. The algorithm is robust as it excludes such statistically or NLP expensive techniques as combinatorial computations or proper tagging and parsing.

Our evaluation results showed up to 98% correctness and high processing speed. It takes a fraction of a second to process a patent on a regular Hewlett-Packard X86-based PC. An XML file of 8 megabytes with150 patents is processed in less than 2 minutes.

We have conducted a feasibility study on porting KeyPat to the French language with promising results. We are now carrying out further experiments investigating ways of KeyPat application to MT analysis stage and automatic multilingual lexicon acquisition.

In this paper we focused on translation-related applications of KeyPat, though, of course, it can directly be used for databases indexing, unilingual and multilingual information retrieval, extraction, summarization and the like.

References

Daille Béatrice and Emmanuel Morin. 2008. A effective compositional model for lexical alignment. *IJCNLP 2008: Third International Joint Conference on Natural Language Processing*, January 7-12, 2008, Hyderabad, India; pp 95-102.

Duda, R.O, and Hart, P.E. 1973. Pattern Classification and Scene Analysis. New York: John Wiley & Sons.

Fattori, Michele., Pedrazzi Giorgio., Turra, Roberta. 2003. Text mining applied to patent mapping:a practical business case. World Patent Information,25.

Hewavitharana Sanjika, Alon Lavie, and Stephan Vogel. 2007. Experiments with a noun-phrase driven statistical machine translation system. *MT Summit XI*, 10-14 September 2007, Copenhagen, Denmark. *Proceedings*; pp.247-253.

Hehenberger M, Coupet P. 1998. Text mining applied to patent analysis. Annual Meeting of American Intellectual Property Law Association (AIPLA), Arlington, VA.

Laporte, Eric, Takuya Nakamura and Stavroula Voyatzi. 2008. A French Corpus Annotated for Multiword Nouns. Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008) pp.27-30.

Martínez-Fernández, J. L., García-Serrano, A., Martínez, P., Villena, J. *Automatic Keyword Extraction for News Finder http://canada.esat.kuleuven.ac.be/omnipaper/downloads/WP7_AKE_AMR_1.pdf*

Matsuo, Y., Ishizuka M. 2003. Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information. International Journal on Artificial Intelligence Tools

Piao, S. L., Rayson, P., Archer, D. and McEnery, T. 2005. Comparing and Combining A Semantic Tagger and A Statistical Tool for MWE Extraction. *Computer Speech & Language* Volume 19, Issue 4, pp. 378-397.

Sheremetyeva S. 2009. On Extracting Multiword NP Terminology for MT. *Proceedings of the EAMT Conference.* Barcelona, Spain, May 13-15.

Tseng Yuen-Hsien., Wang, Yeong-Ming., Juang, Dai-Wei., Lin, Chi-Jen. 2005 *Text mining for patent map analysis.* IACIS Pacific.