Patent Classification for FUSE

Marc Verhagen

Version 4 — September 10, 2012

1 Introduction

The genre classification proposed by Brandeis uses both a flat list and the notion of facets. Some of the genres in the flat classification need to be subdivided, most notably the research-article and patent genres. It seems possible that a more worked-out notion of facets or the (as of yet rather immature) slot labeling approach can help with research-article, but this is unlikely for the patent genre.

This document describes an alternative way to classify patents, one not based on facets but on the notion of technology maturity. Of course, this could be deemed a facet, but that does look rather forced because facets tend to be expressed linguistically.

2 Patent Classification

We use a bi-dimensional classification that is based on both the basic patent type and the maturity of technologies that are referred to in the patent and that are considered prior work. The type is derived from the often quoted classification by the U.S. Patent and Trademark Office (PTO), which uses the following classes:

- Utility Patent. Utility patents are grouped in five categories: a process, a machine, a manufacture, a composition of matter, or an improvement of an existing idea. Often, an invention will fall into more than one of the categories.
- Design Patent. Granted for product designs.
- Plant Patent.
- Re-issue Patent.
- Defensive Publication.
- Statutory Invention Registration

The distinction between these five top types may be useful, but in our data set almost all patents are utility patents. We expect the sub classification within utility patents to be of more use and concentrate our efforts on deriving those subtypes.

For now, we are not exploring other classifications and ontologies. For example, there is a classification of claim types[1], which include Beauregard claim, Exhausted combination, Jepson claim, Markush claim, Means-plus-function, Product-by-process, Programmed computer, Reach-through, Signal claim, and, my favourite, the Swiss type claim. In addition, the PTO has an alphabetical list of subject headings referring to specific classes and subclasses of the classification system (400 subject classes and 115,000 subject subclasses).[2] It is unclear whether these claim types and classes are useful.

2.1 Technology Maturity

In the context of evaluation of retrieval systems on patents, Kando (1999) suggested that the target levels of the technologies evaluated need to be taken into account.[3] While it is not entirely clear to me what Kando is suggesting, it appears to be potentially useful to classify patents along some notion of availability of technologies required by the patent. Kando suggested four types of technology:

1. technology which is readily available on the operational systems;
2. technology which has almost been achieved on research systems but there may be some points to be improved
3. technology that is attractive for users but has not been achieved even on research systems and some promising techniques or models usable for the purpose were known
4. technology that is attractive for users but is quite challenging for implementation

A classification of this type would tell us the stages of the technology and will be a strong indicator for the 'practical application' and the 'commercial application' challenge questions, and therefore deserves further attention. We can couch this in terms of maturity of technologies and their place in the technology life cycle.

Note that the maturity level of technologies has to be determined for all time slices in the document set. That is, instead of simply stating that a technology is of a certain maturity level, we need to store the year when a technology reached that level. We are not sure whether we will be able to have the 4 levels mentioned above, instead, we may make this a binary classification.

[1] http://en.wikipedia.org/wiki/List_of_patent_claim_types

[2] http://www.uspto.gov/web/patents/classification/uspcindex/indexa.htm

[3] Kando, N. (1999). What Shall We Evaluate? Preliminary Discussion for the NTCIR Patent IR Challenge (PIC) Based on the Brainstorming with the Specialized Intermediaries in Patent Searching and Patent Attorneys. http://research.nii.ac.jp/ntcir/sigir2000ws/sigirprws-kando.pdf.

3 Relevance to FUSE

The relevance of document genres to the central FUSE question on emergence is hypothesized to be the following:

> The kind of community of actors that is indicative of an emergent concept is reflected in the literature in many ways. One way is the overall genre mix of publications in on the concept. That is, the distribution over genres is different for an emergent concept as compared to a random set of documents or a publications on a concept that is not emergent.

One defect of the current list of genres is that their distribution over the data is very skewed, with two genres, research-article and patent, dominating the corpus. A further subdivision of these two genres may help the overall cause. In addition, and particular to the technology maturity, we hypothesize that average maturity of technologies referenced in a patent are an indicator of emergence in an RDG. This is illustrated in the figure below, where an increase in maturity scores in 90-95 may indicate a field ready to take off.

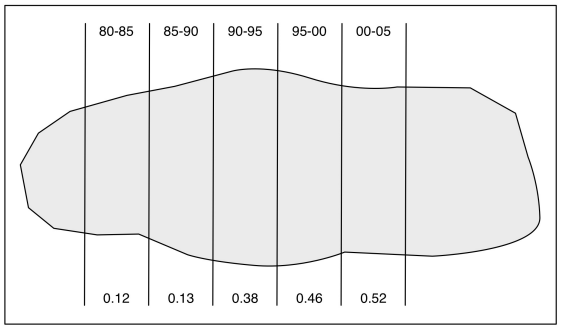

Figure 1: Maturity Scores over Slices of an RDG

All this is to be empirically proven, and the risk is that such a correlation does not exist. Initial experiments on genre distribution are inconclusive due to small sample sizes. No experiments on the maturity scores have been conducted yet.

4 Approach

So patent classification is orthogonal and based on the type of the patent and the maturity of technologies that are referred to in the patent and that are considered prior work. Patent types are as listed in the previous section. For the second dimension, we focus on technologies referenced in the patent and their position in the technology life cycle, that is the maturity of these technologies. Each of the referenced technologies will be associated with an availability score, which uses a Likert scale ranging from mature (the technology is readily available on production systems) to immature (the technology is not available or only available on experimental prototype systems). The patents themselves will receive

a score for technology-maturity, using the same scale and computed from aggregate maturity scores of all technologies. The rest of this section details the patent classification approach.

4.1 Determining the Patent Type

Type detection uses a meta tag as well as a couple of simple heuristics that are applied to specific sections of the patent (see Figure 2).

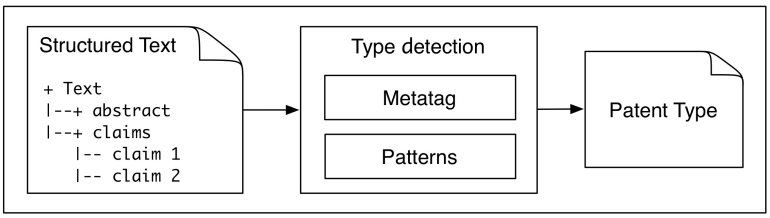

Figure 2: The Patent Classification Module

For English, the top level types appear to be readily available in the metadata of the patent. For example, in patents provided by 1790 Analytics, the top of the xml file looks as follows:

```
<start>
  <US07962427>
    <us-bibliographic-data-grant>
      <publication-reference>
        <document-id>...</document-id>
      </publication-reference>
      <application-reference appl-type="utility">
       <document-id>...</document-id>
```

The `appl-type` attribute encodes the top-level type. More interesting to us is the subdivision underneath the utility type. Simple pattern matching on the abstract of the patent and the claims list (especially the first claim) can provide this information. For example, for patent US07962427, the beginning of the abstract is *A method and system for determining whether...* and the first claim starts with *A method of determining that...*, which indicates that US07962427 patents a method (or process).

It may be interesting to also look at other independent claims as well as at dependent claims in the patent. And it may be useful to not simply classify a patent as process or machine, but instead give weights to all possible categories. Here are some observations on patent US08027663, with the following abstract:

> A wireless device includes a data capture system, a radiant-energy data transmission system, and a steganographic encoder that hides a plural-bit auxiliary code within data captured by the data capture system prior to its transmission by the data transmission system. An illustrative system, operable with audio input data, is a cell phone that steganographically encodes a user's voice. In

some arrangements the steganographic encoding depends on data wirelessly received from a remote location. A variety of other arrangements and systems are also detailed.

The abstract mostly talks about devices and systems, the first 5 claims are all about methods, but claims 6 though 8 are all on devices and mediums, for example:

```
5. A method of steganography usage in a wireless...
6. A tangible computer-readable medium having...
7. A wireless communications device comprising...
```

For a single example taken from a patent on semiconductor chips, the claim types are (this example is taken from an email from Patrick Thomas):

1. Composition of matter a new compound or formulation, for example a new material for a semiconductor chip
2. Process (or method) a new method of doing something, for example a new method for etching a chip
3. Device an article that is produced, for example a chip itself (the device category is often divided into Machines, which have moving, interacting parts, such as a sewing machine; and Articles of Manufacture, which contain static parts, such as a shovel).

Another option would be to drop the fixed sub classification of utility patents (process, machine, manufacture, composition of matter, improvement of an existing idea) and instead take the lexical items that are found in the claims (method, device, medium).

Inspection of Chinese and German patent showed several things:

1. There is no clear distinction between the five top-level types
2. Differences between for example processes and devices are not marked lexically. In fact, this also turned out to be true for a significant subset (no numbers available yet) of English patents.

As a result of this, classification along the process-device distinction, while feasible for a lot of English documents seems much harder for Chinese and German and would probably require a significantly large vocabulary of terms classified along said distinction. For these reasons, it was decided that type detection was not going to be an element of the Chinese and German components.

4.2 Determining the Technology Maturity Score

Ideas on how to design the system to compute this scores have evolved considerably over the last few weeks (see version 2 of this document). Technology maturity scores are relevant at three levels:

1. Technology Level — Each technology can be assigned a score reflecting its maturity, we are using a three point Likert scale with the values *not available*, *immature*, and *mature*. These will often be represented with the integers 0, 1 and 2 respectively.

2. Patent Level — A patent receives an aggregate score based on technologies referenced in the patent and the maturity scores associated with those technologies. The score will be on a scale from 0 to 1.

3. RDG Level — The maturity score of the RDG is simply the average of the maturity scores of all patents in the RDG. In addition, scores for various time slices of the RDG are computed as well.

The system has three main components: the pattern generator, the ontology builder and the runtime system. See the image below for an overview of the architecture.

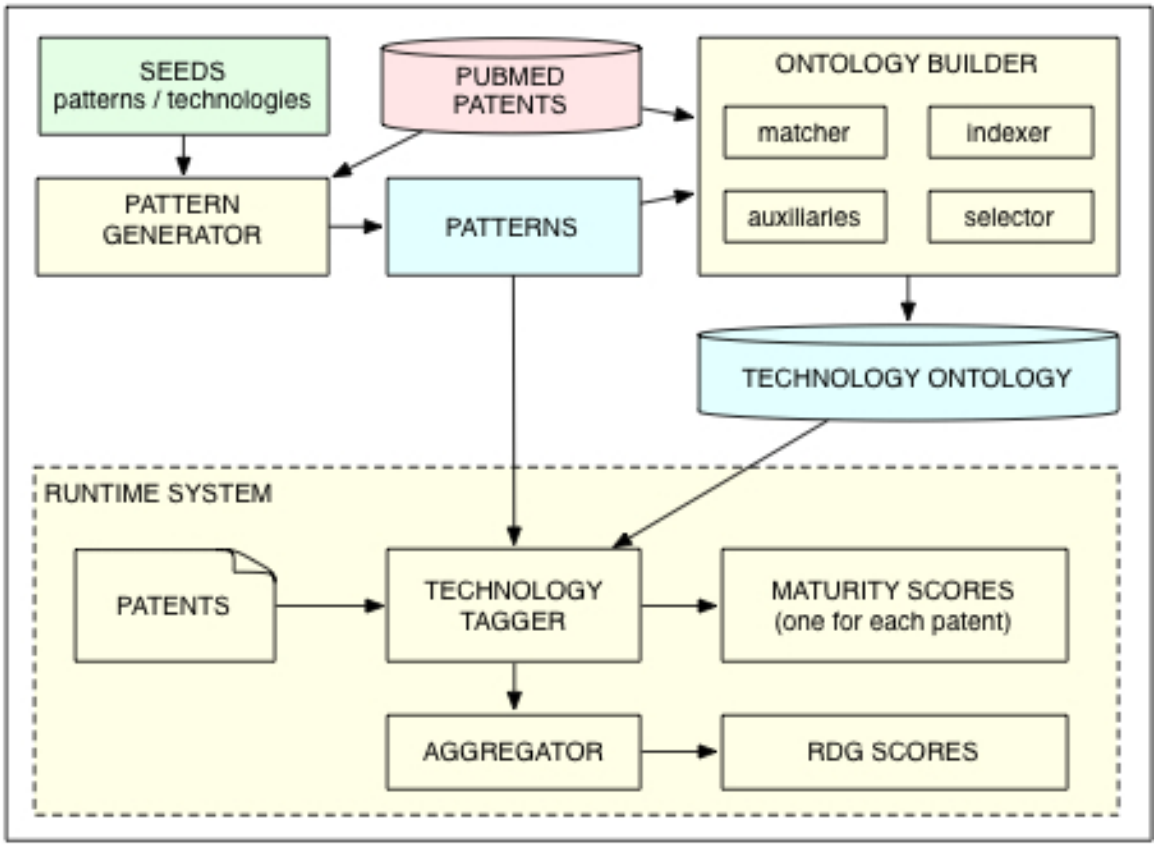

Figure 3: The System for Generating Maturity Scores

The Pattern Generator generates patterns for each language and domain, given a set of seed patterns and seed technologies and a large set of training data. The Ontology Builder takes these patterns (actually, a subset thereof) and creates the Technology Ontology, which is used in the runtime system by the Technology Tagger We now discuss these components in more detail.

4.2.1 Generating the Patterns

The current system is geared towards minimizing manual labour when generating the patterns and is conceptually split up in two stages. The first stage is the generation of the bare patterns (see Figure 4).

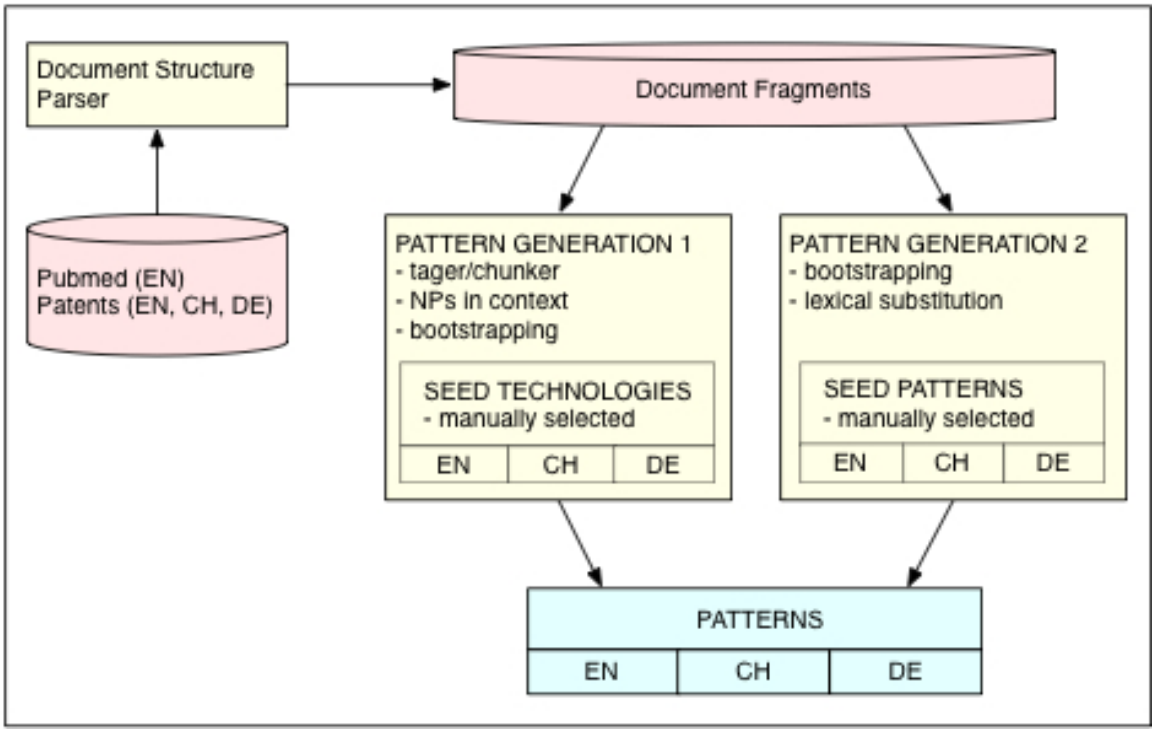

Figure 4: The Two Components for Generating Patterns

There are three inputs to the Pattern Generator: seed patterns, seed technologies and the Pubmed and LexisNexis source data. Originally, we also intended to use the output of the NYU terminology tagger in this phase. The NUY terminology extractor does not tag individual technologies in a document but gives a ranked list of terms for a document set. We intended to use this list to reduce search space and constrain to what sentences patterns were applied, leaving it is an empirical question to decide whether this initial list is needed or whether this list reduces the search space too much. The list inspected contains 3146 terms, extracted from scientific articles on DNA. The list contains a large number of more general nouns in the lower 30% of the list. More importantly, the list seems to contain very few instances of clear technologies. For example, the first 100 lines only contained a few terms that could clearly be classified as technologies. This may force us to amend the list using resources like the UMLS thesaurus or other domain ontologies to increase the size of the initial term list.

For creating the technology list from the term list, we originally intended to use a domain expert to pare down the list and for each technology to list the year when it became available. This idea was deserted for three reasons: (i) concerns that the resulting ontology may be too small to be useful, (ii) unclarities on how to actually define technology and

potential time-consuming complexities in determining the year, and (iii) concerns that this approach does not scale well because the same manual procedure would have to be done for each new subject domain.

So we will not use the terminology list at this stage, but we may use it in the Ontology Builder as one of a series of heuristics to determine whether a term qualifies as a technology (see Section 4.2.2).

The pattern set is intended to include patterns that provide textual clues to determine availability of technologies or their maturity level, for example:

(1) `TERM became widely available in YEAR`
`TERM, an experimental approach to|for`
`We used TERM to create X`

Different kinds of evidence are searched for. The first pattern matches general statements that provides global information, namely that TERM is a technology that became widely available in YEAR. The second pattern is similar in that TERM is branded available yet only recently (and that it is probably not widespread), the difference is that the publication time of the document that contained the match has to be taken into account. Phrases that match the third pattern are prove that there is an instance that TERM was used for something, again, the year must be taken from the publication time of the document.

Note that these patterns can be considered operational definition of technologies as "terms used in certain contexts". Also note that filters are needed to weed out unwanted results from these patterns. For example, the third pattern above will match with the phrase *we used color to create an impression of speed* can hardly be an indication that *color* is an available technology. We will get back to this issue in later sections. Also note that patterns may be different for scientific literature versus patents.

In addition to the patterns above, there will also be patterns that can be used simply to determine that a term is a patent, without shedding any light on whether, when and how often these technologies were used.

`SOME EXAMPLES TO BE ADDED`

There are basically two approaches to generating patterns. Both run on document fragments where the fragments are selected using the output of the Document Structure Parser. Sentences are extracted from a limited set of sections (starting with just the content of the abstract, but adding other sections when it seems appropriate) and stored off-line in an intermediate data structure. The generation approaches differ mainly in terms of what kind of seeds they take:

1. The first approach uses seed technologies. Sentences are chunked and considered as NPs in context and some of the NPs are marked as technologies. Simple bootstrapping techniques are used to determine contexts that occur often next to technologies and these contexts are used as patterns as well as vehicles to generate more technologies.

2. The second approach starts off with seed patterns and uses lexical substitution and bootstrapping to generate more patterns.

All these two approaches do is generate a set of base patterns. As mentioned above, many of these patterns overgenerate by a large margin so at least some measure of how reliably these patterns indicate the existence of a technology has to be added. In addition, the patterns are not classified according to the types shown above. This is where the second stage of pattern generation comes in (Figure 5).

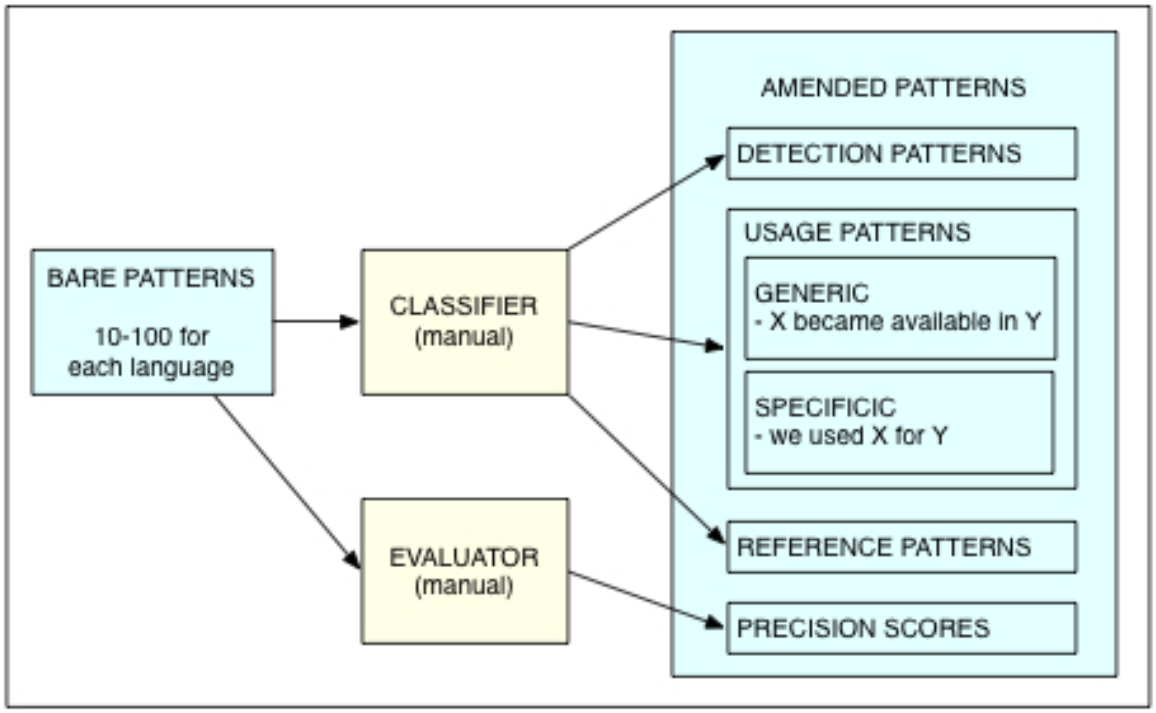

Figure 5: Classifying and Evaluating the Patterns

This phase is entirely manual but the amount of labour is constrained to a manageable task as explained below. First we run the patterns on a subset of the data, collecting matching contexts. Then, on the evaluation end, we collect a hundred matches each pattern and determine manually whether the result did indeed contain a technology in its context. Matches are given one of three scores: (i) the result clearly contains a technology, (ii) the results does clearly not contain a technology and (iii) the result contains something that looks like an ontology. We define technology loosely as something that can be used in scientific research as a resource that helps further the goal of the research. This could refer to hard technologies like control rods, micro processors and other physical technologies, medications, drugs and treatments (5-fluorouracil, adjuvant chemotherapy), and medical technologies (mirna target site, lambda dna replication, hairpin probe). It also includes methods, processes, theories and algorithms (random matrix theory, support vector machines). Each pattern is associated with a precision score ranging from 0 to 1 which indicates how reliably the patterns extracts technologies.

This process is restrained in several ways. First, we do not allow the Pattern Generator to generate more than a few hundred patters. Second, we can first use the above procedure on all patterns using only 10 matches and then feeding only the top 50 of those patents into

the phase where we use 100 matches. All other patterns are discarded.

The remaining patterns are then manually classified according to the types listed in the beginning of this section and repeated in Figure 5. It is allowed for patterns to be in more than one class. Reference Patterns are explained in more detail in Section 4.2.3.

4.2.2 Creating the Technology Ontology

With the patterns in place, the ontology can be built using these patterns in addition to a set of auxiliary methods that serve to add more evidence, positive or negative, as to whether a technology recognized by patterns actually is a technology.

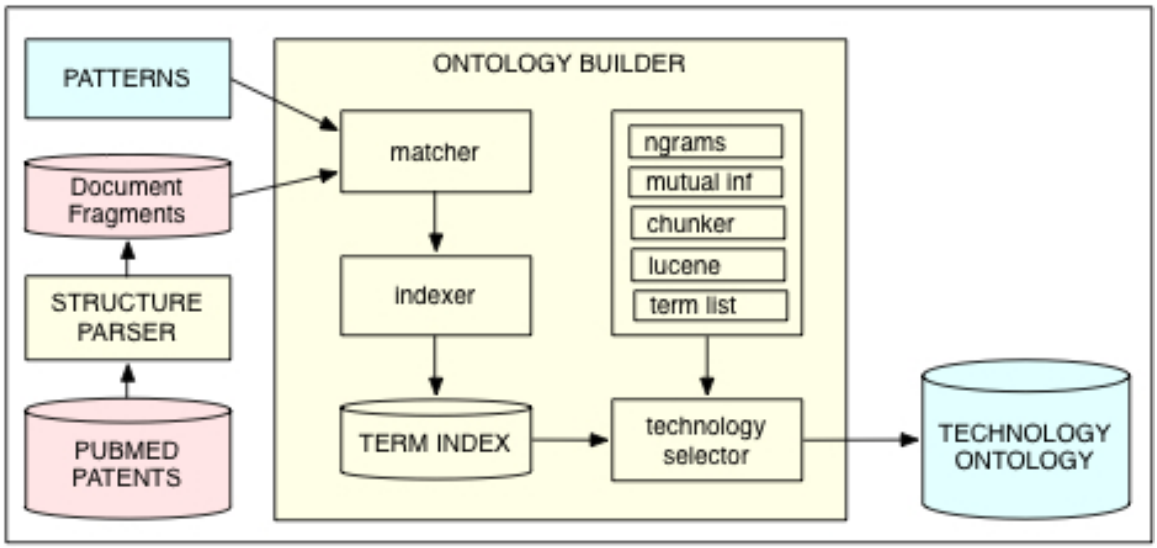

Figure 6: The Ontology Builder

First the matchers takes all patterns and applies them to fragments selected from the source data by the Document Structure Parser. Sources used are Pubmed for English, and LexisNexis patents for English, Chinese and German). Next, the indexer takes the list of matches and creates an inverted index where terms are mapped to the set of patterns that matched. The example below shows how an inverted index is created from a set of matches associated with a single pattern.

(2) a. PATTERN: We used TERM to create X

b. MATCH: We used $term_i$ to create X
MATCH: We used $term_i$ to create Y
MATCH: We used $term_j$ to create Z

c. $term_i$: {We used $term_i$ to create X, We used $term_i$ to create Y}
$term_j$: {We used $term_j$ to create Z}

This listing is by no means complete however. For housekeeping purposes, some more information is included in the index. The list of matches is structured by pattern and

perhaps by pattern type. And for each pattern, the index stores, or has pointers to, general information on the pattern like the precision scores and pattern class. In addition, for each match some information is stored on where the match occurred, including document name, publication date of the document, the document section, and the sentence that matched. Note that these could all be pointers to the source.

Conceptually, what is happening is that we collect pieces of evidence and create a matrix by plotting whether terms occur in contexts specified in patterns. All cells in the matrix are initialized with 0. If we find an occurrence of this pattern for term t_i in a patent from year y_j, then we increment the integer at Matrix(t_i,y_j). This would result in a matrix like the one below.

	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
t1	0	1	0	0	0	0	1	3	2	0	0	2	0	4	6	5	8	8	7	7
t2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t4	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0
t5	0	0	0	0	0	0	0	0	0	0	0	1	0	1	2	1	0	2	3	2

In addition to instances of technologies being used, the matrix would also store general statements matching patterns like the first two in example (1). So each cell would not just contain the one number as above, but also a list of general statements. It is an open issue how to combine these two sources into one generalized number. From this matrix we build a technology ontology. First we remove terms for which the following two conditions hold:

1. the sum of the numbers in a row is below N
2. the number of general statements in the row is below M

The values of M and N are to be determined but the simplest case would be to set them to 1 and throw out all rows that are all zeros, which would get rid of t2 and t3 (assuming there are no general statements in those rows). With slightly higher thresholds we throw out rows with low counts and t4 would be a candidate here. The remaining terms will be put in the technology ontology and will be associated with a time span that they were available and a time span that they were not, smoothing out the numbers in the matrix above, while taking into account the general statements.

	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
t1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

This is using a binary scale, but it may be possible to have a Likert scale with more values, in which case we would see monotonic increases in the numbers in a row. An research question that we may want to explore is whether given training data from a time slice like 1980-1984 we can predict when technologies t1 and t5 will become widely available.

The story above is simplified in the sense that it abstracted away from what actually is in the term that matched the pattern. The Technology Selector uses additional heuristics that both enrich and expand the index:

1. Index entries are added by taking ngrams of the patterns context
2. Mutual information scores are associated with all terms
3. terms are looked up in Lucene to add time stamped counts
4. a chunker be be used to find noun groups and or noun phrases
5. the term list from NYU may be used to mark index entries that occur in the list

This list may be expanded with other modules. The technology Selector takes all these into account when deciding whether a term is actually an ontology.

4.2.3 The Runtime System

Most processing is done off-line when creating the pattern set and building the ontology. The runtime system is kept lean and fast and the core of what it does is to look up what technologies occur in the patent and calculate maturity scores for each patent, given the maturity scores of the technologies as listed in the ontology (see Figure 7).

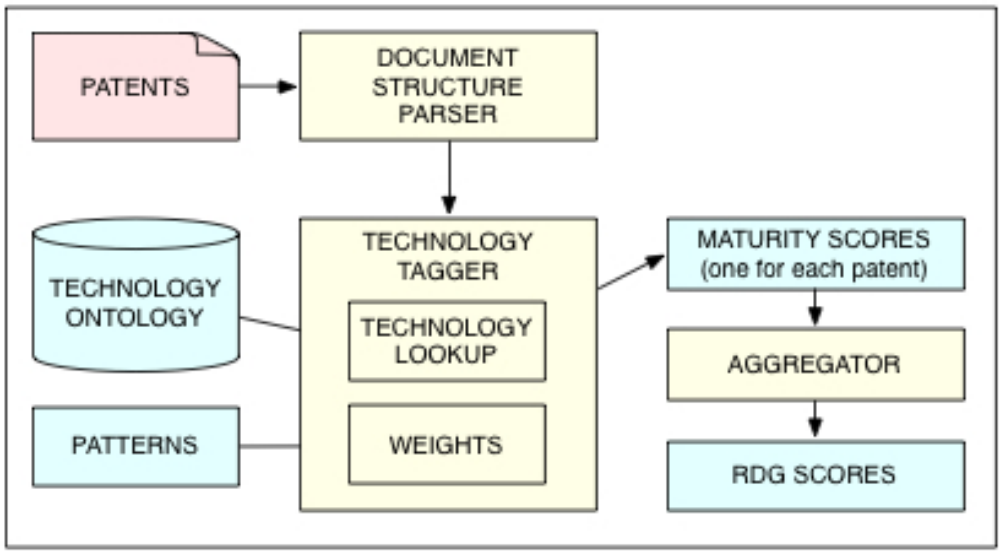

Figure 7: The Technology Tagger in the Runtime System

Extracting Technologies In its barest form this is simply a matter of string matching against the technology ontology. We will however make this process sensitive to document structure by filtering out technologies that do not occur in certain sections like Prior Work, or by assigning different weights according to where technologies are found. Our focus is not primarily on what technologies an individual patent depends on, but on the cloud of technologies that are referenced in a cohesive set of documents (that is, a RDG), and the approach below reflects this. However, for future research we do want to explore the issue of relevance and we would use specialized patterns for that. For each technology, the maturity level is looked up using the year when the patents was issued.

Calculating the Technology Availability Score This is a weighted average of all individual technology maturity scores. Technology scores are weighted only if weights were used in the previous step. The resulting number is on a scale from 0 to 1.

RDG-level Scores The Aggregator simply takes the scores for all patents and generates the average. In addition, averages are computed for all years and for all time slices of three and five years.

5 Application to new Domains

NOTE: THIS SECTION NEEDS TO BE UPDATED BUT I LEFT IT IN AS IS FOR NOW

With the domain expert taken out of the loop in the ontology creation phase, application to new domains is mostly automatic. This is assuming that terminology extractions works across domains. There are however two manual steps:

1. Creating the seed patterns and supervising the pattern inducer
2. Enhancing the term list with ontology data

Both these steps should be relatively straightforward. Given that the current terminology list is built for scientific articles in the biomed domain, we should apply the procedure to at least biomed patents and some other domain in the Elsevier data.

6 Schedule

NOTE: THE SCHEDULE NEEDS TO BE UPDATED AND INDIVIDUAL TASKS MATCHING THE DIAGRAMS ABOVE NEED TO E ADDED.

Jul 2012	Sample input files and output files for English
Aug 2012	Sample input and output files for Chinese and German
Aug 2012	Assessment-specific user manual for genre classifier and list of technologies for English
Sept 2012	List of technologies for Chinese and German
Sep 2012	Dry run