Search for Patents Using Treatment and Causal Relationships

Ashwathi Krishnan
Computer Science Department
University of California, Los Angeles
Los Angeles, California 90024
ashwathi@cs.ucla.edu

Alfonso F. Cardenas
Computer Science Department
University of California, Los Angeles
Los Angeles, California 90024
cardenas@cs.ucla.edu

Derek Springer
Computer Science Department
University of California, Los Angeles
Los Angeles, California 90024
derekspringer@gmail.com

ABSTRACT

An interesting area of research in information retrieval is that of relationship extraction. The ability to scan an article or set of articles and extract relationships such as “X treats Y” or “A happens because of B” is key to retrieving articles of interest to a large population.

In this paper, we describe our method of identifying and extracting treatment and causal relationships from medical patent documents. We use a medical patent corpus to show that using relationship patterns to retrieve medical patent documents helps improving the recall of the system immensely. We also show that expanding our search to look for a broader set of relationships and including causal relationships along with treatment relationships, addresses a larger range of patent documents thereby improving the recall of the system significantly.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *Linguistic processing.*

General Terms

Experimentation, Languages

Keywords

Patent retrieval, Treatment relationships, Causal relationships

1. INTRODUCTION

The presence of relationship information in articles can be used as one of the basis for retrieving articles according to the user’s interest. However, relationships in articles are often highly domain dependent. For example, in the medical domain there are two common relationships, which are also the focus of our research: ‘causal’ and ‘treatment’ relationships. An example of a ‘causal’ relationship may be “A is caused by B”; while a ‘treatment’ relationship is similar to the construct “A is used for treating B”.

We primarily focus on describing a system that we have developed which can automatically detect and locate treatment and causal relationships from medical documents. Such a system would find applications in various areas within the medical domain including (but not limited to) semantic searching of drug patent documents and querying medical journals for causes and cures to common diseases. Our system can be used as an efficient way to search these documents for relationships instead of using the conventional keyword-based searches. We hypothesize that our approach is also useful in finding prior art related to the type of relationship that we address herein.

Section 2 reviews prior work done in the area of patent and relationship search. Section 3 describes our hypothesis and how we propose to use the relationship extraction system that we have developed to discover the relationship patterns in text. We give a brief overview of the relationship extraction system used in Section 4. The algorithm implemented in our system, the experiments we used to test it and their results are discussed in Section 5. Lastly, we draw various conclusions from the results achieved and present them in Section 6.

2. BACKGROUND

The demand for a powerful patent search system and an effective information retrieval method for patent documents are growing with the number of patents steadily increasing all over the world. [8] presents a patent search and classification system by dividing the patents into several collections according to the area dealt with in the patent document. It uses a tf-idf scoring to retrieve and rank patent documents. [9] examines a method for organizing collections of patent documents by topic as well as ranking and selecting them based on these topics.

Research in the field of retrieving patent documents using semantic search techniques has been increasing. There has also been a lot of study in trying to retrieve semantic information from patents. [10] discusses a supervised learning algorithm to extract useful information using regular expressions. Specifically, for patent documents it generates expressions that match and identify the problems solved in the patents. [11] introduces a patent document processing system called PATExpert, which provides an integrated environment for storing, viewing, and searching patents. This system proposes a content representation schema for document patents and suggests two different techniques to process the patents using this schema. It is based on the recent ontology technology.

Content-based semantic search strives to improve search accuracy by understanding searcher intent and the contextual meaning of terms as they appear in the search space to generate more relevant results. This method of searching when used with patent retrieval as opposed to keyword-based search is more powerful in retrieving relevant patent documents. This has been extended to

PaIR’10, October 26, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0384-2/10/10...$10.00.

patent image retrieval as well. [12] and [13] propose two such systems. An image based search system called PATSEEK is detailed in [12], which uses similarity retrieval concepts to search patent documents using query images. [13] introduces indexing and image analysis techniques for patent image search and retrieval systems.

Roxana Girju at Baylor University proposes a novel and innovative method [1] to automatically detect and extract causal relations from text. Additionally, the method is extended to automatically discover lexical and semantic constraints necessary for the disambiguation of causal relations, which is, then used in question answering. Chu [4] has adapted this method by using lexico-syntactic patterns of the form NP1 VP1 NP2 (where NP means noun phrase and VP means verb phrase) to identify treatment relationships where the VP contains the treatment pattern and the two NPs contain the subject and the object. The system uses a three step process in extracting treatment relationships which involves using pre-defined lexico-syntactic patterns to identify and extract treatment relationships (specifically treatment verbs), manually tagging the list of returned relationships to build up a training corpus and, finally, learning the classification rules to determine a valid relationship using a statistical classifier on the training corpus.

The Espresso algorithm [3] uses a different approach to identify semantic relations. With minimal supervision, it uses generic patterns to identify the relations and measures for pattern and instance reliability to filter out the incorrect patterns. The algorithm substantially increases system recall with small effect on overall precision. In this paper, we adapt the Espresso system to extract treatment and causal relationships and augment the system with a larger set of binary relationships for the purpose of a semantic patent search engine.

Table 1. Taxonomy of Binary relationships

Relative Frequency	Category	Simplified Lexico-Syntactic Pattern
37.8	Verb	E_1 Verb E_2 *X established Y*
22.8	Noun + Prep	E_1 NP Prep E_2 *X settlement with Y*
16.0	Verb + Prep	E_1 Verb Prep E_2 *X moved to Y*
9.4	Infinitive	E_1 to Verb E_2 *X plans to acquire Y*
5.2	Modifier	E_1 Verb E_2 Noun *X is Y winner*
1.8	Coordinate$_n$	E_1 (and|,|-|:) E_2 NP *X-Y deal*
1.0	Coordinate$_v$	E_1 (and|,) E_2 Verb *X, Y merge*
0.8	Appositive	E_1 NP (:|,)? E_2 *X hometown: Y*

3. HYPOTHESIS

Relationship extraction is the task of recognizing the assertion of a particular relationship between two or more entities in text. Banko [2] claims that 95% of all binary relations found using a sample set of 500 random sentences belong to one of the categories listed in Table 1. Chu's approach [4] uses lexico-syntactic patterns of the form NP1 VP NP2 (Verb category listed in Table 1) to identify treatment relationships. The VP would contain the treatment verb or pattern and the two NPs would contain the subject and object. This structure is a very common relationship structure as evidenced by Banko [2] (in Table 1) where the pattern E1 Verb E2 (E1 and E2 denote subject and object) accounts for 37.8% of all relationships. However, there still remain a large number of worthwhile relationships that may provide fruitful results. We explore these by expanding our implementation to include other relationship categories shown in Table 1 – Noun + Prep e.g. "X settlement with Y", Verb + Prep e.g. "X moved to Y", Infinitive e.g. "X plans to acquire Y", and Modifier e.g. "X is Y winner". We hypothesize that the usefulness of the system increases greatly when a larger number of relationships are addressed i.e. 91.2%, as compared to the previous figure of 37.8%, thereby improving the recall of the system.

Treatment relationships refer to any case where A (the subject of the relationship) can be used in the treatment of B (the object of the relationship) to lessen the adverse affects of B. In most of the cases, B will be some sort of negative disease or condition state such as depression, arthritis or fever. On the other hand, A may be a drug such as Tylenol, an activity such as surgery, or something else that can be used to treat B. Causal relationships refer to any case where B is caused by the condition C. In such situations, B is again a negative disease or condition state whereas C is a state of the body or environment which brings about the disease B. Our hypothesis is that many users will also be interested in the causality aspect of disease or condition B. To this end, we develop our system to identify both treatment and causal relationships and extract them from the drug patent corpus.

Finally, we speculate the possibility of retrieving a larger set of relationships by searching for the synonyms of the subject and object in each relationship, and adding other relationships, in which they occur, to the pool of relationships to be examined. Using these retrieved relationships, the treatment and causal verbs we will obtain from section 4.2 are examined similarly and their synonyms are obtained. Furthermore, to these we also add the various degrees or modulations of the verbs with relevance to the medical domain. We believe that augmenting the system with different variations of the extracted treatment and causal verbs would also facilitate in achieving a higher recall. We hypothesize that our approach is also useful in finding prior art related to the type of relationship that we address herein.

4. SYSTEM OVERVIEW

4.1 The Espresso Algorithm

Espresso [3] is a general-purpose, broad, and accurate corpus-harvesting algorithm that requires minimal supervision. It proposes a novel method for exploiting generic patterns, especially patterns with high recall and low precision. Unlike previous algorithms that required significant manual work to make use of generic patterns, this work proposes a novel filtering method for using generic patterns. Additionally, it proposes a new measure of pattern and instance reliability that enables the use of generic patterns.

Espresso is a minimally supervised bootstrapping algorithm that takes as input a few seed instances of a particular relation and iteratively learns surface patterns to extract more instances. To that effect, Espresso iterates between the following three phases: pattern induction, pattern ranking and instance extraction. The algorithm begins with seed instances of a particular binary relation (e.g. is-a, causal, etc.) and then iterates through the phases until it extracts a certain fixed number of patterns or the average pattern score decreases by more than 50% from the previous iteration.

In the pattern induction phase, Espresso infers a set of surface patterns P that connect as many of the seed instances as possible in a given corpus. Any pattern learning algorithm can be used for this purpose but the authors of the Espresso algorithm chose the algorithm described in [5]. After Espresso extracts patterns for all the given seed instances, it then ranks the patterns in P according to the reliability measure r_{Π} (discussed in Section 4.3) and disregards all but the top-k patterns where *k* is set to the number of patterns from the previous iteration plus one. In the instance extraction phase, Espresso retrieves from the corpus, the set of instances *i* that match any of the patterns in P. Then, a principled measure of reliability r_i is calculated for each instance. Espresso then filters out incorrect instances and selects the highest scoring instances as input for the subsequent iteration. For our relationship extraction system, however, we use an adaptation of the Espresso algorithm that is detailed in the following sections.

4.2 Pattern and Instance Reliability

A reliable pattern is one that is both highly precise and one that extracts many instances. The recall of a pattern *p* can be approximated by the fraction of input instances that are extracted by *p*. Since it is non-trivial to estimate automatically the precision of a pattern, keeping patterns that generate many instances might not be a good idea (i.e., patterns that generate high recall but potentially disastrous precision). Hence, patterns that are highly associated with the input instances are desired. Point-wise mutual information (pmi) is a commonly used metric for measuring this strength of association between two events *x* and *y*:

$$pmi(x,y) = \log \frac{P(x,y)}{P(x)P(y)}$$

The reliability of a pattern p, $r_{\Pi}(p)$*,* is defined as its average strength of association across each input instance *i* in the set of instances *I,* weighted by the reliability of each instance *i*:

$$r_{\Pi}(p) = \frac{\sum_{i \in I} \left(\frac{pmi(i,p)}{\max_{pmi}} * r_i(i) \right)}{|I|}$$

Where $r_i(i)$ is the reliability of instance *i* and max_{pmi} is the maximum point-wise mutual information between all patterns and all instances. The reliability of the manually supplied seed instances is 1. The point-wise mutual information between instance $i = \{x,y\}$ and pattern *p* is estimated using the formula:

$$pmi(i,p) = \log \frac{|x,p,y|}{|x,*,y||*,p,*|}$$

Where $|x, p, y|$ is the frequency of pattern *p* instantiated with terms *x* and *y* and where the * represents a wild card. Estimating the reliability of an instance is similar to estimating the reliability of a pattern. A reliable instance is one that is highly associated with as many reliable patterns as possible. Hence, it is analogous to the pattern reliability measure:

$$r_i(i) = \frac{\sum_{p \in P} \left(\frac{pmi(i,p)}{\max_{pmi}} * r_{\Pi}(p) \right)}{|P|}$$

Where $r_{\Pi}(p)$ is the reliability of pattern *p* and max_{pmi} is as defined before. Thus, $r_{\Pi}(p)$ and $r_i(i)$ are recursively defined.

4.3 Extracting Verbs

The relationship extraction system that we have developed is initially fed with a set of known treatment and/or causal relation instances of the form <subject, object>. The algorithm stores these relations in a map called *relevant_relations*, where each instance is a key and its value denotes its relevancy score (as outlined in [3]). The input instances all have r_i values equal to 1.0. Using the input instances, we then extract verbs from the test corpus that participated in a relationship described in the categories (as discussed in section 3). We obtain the synonyms of the subject and object of the relationship and then use these synonyms in the search as well. The UMLS medical dictionary [5, 6] created by the National Library of Medicine, which contains over 1 million medical terms, is used for this purpose.

After the sentences are collected from the corpus, we process these sentences to be able to easily extract the pattern or verb connecting the subject and object of the relationship. We use the PCFG (Probabilistic Context-Free Grammar) shallow parser [7] to process sentences by reducing words to their base form and assigning part-of-speech tags to each of the words in a sentence. The parser also "chunks" (collects) tagged words together into phrases such as noun phrases, prepositional phrases and verb phrases. Then, we locate the subject and object terms (or their synonyms) of an input relation within the processed sentence and return the verb phrase between them (if it exists). If more than one verb phrase exists between the subject and object, then we exclude all of the verb phrases from that sentence, as the verb phrases would not likely link the subject and object of the treatment relationship. For example, if we have the sentence structure NP1 VP1 NP2 VP2 NP3 with NP1 and NP3 containing the input subject and object, we cannot take into account VP1 and VP2 as possible patterns because both these verb phrases do not link NP1 and NP3 directly (e.g. the architects established plans to build the monument). Finally, we identify and extract specific verbs from the verb phrases returned. We do this because searching for other relationship instances using whole verb phrases is too restrictive. Thus, only individual verbs from phrases are retained i.e. those words that have a tag starting with 'v'. For example, from the verb phrases [VP significantly/r reduce/v] we retain the verb "reduce".

It should be noted that the algorithm to extract relevant verbs from a corpus only takes as input the given seed relations. For any given corpus the extracted verbs will remain the same for any given run or test as long as the input seed relations remain unchanged. The verbs extracted from one corpus will differ from the verbs extracted from a different corpus.

4.4 Extracting Relationship Patterns

Each extracted verb from section 4.3 is then inserted into another map called *relevant_verbs* that has a possible verb as the key and its relevancy score r_{Π} as its value. Initially, all verbs put into the map have an r_{Π} score of 0.0. We then calculate the r_{Π} values of all verbs in the map according to the formula in section 4.2, where

$r_i(i)$ is 1.0 for all the instances we know so far (only the input instances are known at this stage) and $|I|$ is the number of input instances. The algorithm used to compute the $r_\prod$ scores is [3]:

```
For all verbs v in relevant_verbs
{
  For all instances (s, o) in relevant_relations with value $r_i(i)$
  {
          P = pmi (s, o, v)
          $r_\prod(v)$ += P * $r_i(i)$
          Update max_pmi if P is greater than old max_pmi
  }
}
```

The point-wise mutual information (pmi) in our system is calculated between a relationship instance and a verb. Here the relationship instance corresponds to a subject-object pair *(s, o)* and a treatment or causal verb *v*. This is achieved by first querying the data corpus for the count of sentences in which *s, o* and *v* appeared together giving the value for *count_sov*. Similarly, the corpus is queried for the count of sentences in which *s* and *o* occurred together and a count of sentences in which the verb *v* occurred, to give the values *count_so* and *count_v* respectively. As a result, the computation of pmi (as discussed in section 4.2) translates to:

$$pmi(s,o,v) = \log \frac{count\_sov}{count\_so * count\_v}$$

Once the $r_\prod$ values are calculated for all the extracted verbs, the verbs having low $r_\prod$ values are filtered out. This can be done in two different ways:

1. Setting a threshold for the $r_\prod$ values and filtering out those verbs with $r_\prod$ lesser than the threshold. The threshold is set by manually observing all the $r_\prod$ values.
2. Choosing the top *X* number of verbs that have the highest $r_\prod$ values where *X* can be between 10 and 30.

In our experiments, we tried both methods and this process did not affect the results obtained. In general, we tried to control the number of relevant verbs that this stage of the algorithm returned so as to optimize the run time of the rest of the algorithm (more the number of relevant verbs implies more time would be required to process the r_i of each relationship instance).

After the irrelevant verbs from our map are filtered out, we form a test sentence set of those sentences containing one of the relevant verbs. The number of sentences for each relevant verb in our test sentence set varies with the experiments performed as detailed in sections 5.5 and 5.6.

4.5 Extracting Relationships from the Test Sentence Set

Once the test sentence set is constructed, each sentence in the set was processed using the PCFG shallow parser [7]. From these parsed sentences, the noun phrases surrounding the verb phrase, which formed a pattern in one of our target categories, containing a relevant treatment or causal verb are extracted. Additionally, we check for the presence of certain prepositional phrases directly following the verb phrase, which gave us an indication of whether the sentence was active or passive. For example, the presence of a 'with' following the relevant verb in a sentence indicates that it was passive as in the sentence "depression is often treated with Zoloft". In passive sentences, the subject was searched for in noun phrases after the verb and the object was searched for in noun phrases before the verb. The opposite was true in the case of active sentences. For each of the extracted noun phrases, only the actual nouns were kept as part of the relationship. For example, in the noun phrase [NP cancer/n and/c] we dropped the conjunction "and" and retained just the noun "cancer". All the extracted relationships were again stored in the map, relevant_relations, with initial values of 0.0.

In the next stage, we calculated the r_i scores of all the extracted relationship instances according to the formula in section 4.2, where $r_\prod$ *(p)* is the relevancy score calculated for relevant verb *p* (which is stored in the relevant_verbs map) and $|P|$ is the total number of relevant verbs extracted in the previous stage. *pmi(i, p)* is as calculated before. The algorithm to calculate the relevancy score of relationship instances r_i is similar to the algorithm we used to calculate $r_\prod$ of the extracted verbs, and is as given below:

```
For all instances (s, o) in relevant_relations
{
  For all verbs v in relevant_verbs with value $r_\prod(v)$
  {
          P = pmi (s, o, v)
          $r_i(i)$+= P * $r_\prod(v)$
          Update max_pmi if P is greater than old max_pmi
  }
}
```

Once the r_i for all the extracted instances are calculated, we filter out those instances as being incorrect that had r_i values less than the threshold value. For each experiment, we conducted some micro-benchmarks to determine the optimum threshold value. Those instances with r_i score greater than the threshold were determined to be correct by the system and extracted as the relationships from the test sentence set.

The output set of relations is then manually examined and tagged as being correct or not. This enables us to calculate the precision of the system. We use the original sentence from which the relation was extracted as a reference point for determining correctness. Additionally, we also manually tag all the relations that were extracted at the end of section 4.3; including relations whose r_i scores were less than the threshold. Counting the number of correct relations after this tagging procedure gave us a number for the total number of correct relations present in the test sentence set which was then used in calculating the recall of the system.

5. APPROACHES/ALGORITHMS, EXPERIMENTS AND RESULTS

The algorithm that we implemented to extract relationships from medical text is an adaptation of the Espresso algorithm [3]. We adapted the algorithm to specifically extract treatment and causal type relations of the categories Verb, Noun + Prep, Verb + Prep, Infinitive and Modifier (see Table 1) and to, specifically, extract relations from individual sentences in our corpus (as opposed to an entire text). Additionally, we modified the algorithm slightly so that multiple iterations were not made through the extraction process, as it was discovered that multiple iterations took significantly longer with the quality of results obtained being poorer.

5.1 Dataset – Drug Patent Corpus

A medical dataset rich in treatment and causal relationships was used in the experiments to analyze the system developed. This database comprises of 50,000 drug patent documents extracted from Class 424 & 514 of the U.S. Patents Classification: “drug, bio-affecting and treating compositions” and their subclasses. The patents in Class 424 & 514 were pre-filtered and only those documents containing at least one of the keywords “diabetes”, “metastatic”, “cancer”, “tuberculosis”, “lung”, “bronchitis”, “coronary” and “artery” were added to the corpus. Each sentence from every patent document was then added as a separate tuple in a sentence table under the schema. Thus, the corpus has a test set of about 43 million sentences related to medical patent documents.

5.2 Bootstrapping Lexical-syntactic relationships

The underlying framework of Chu’s [4] implemented automatic treatment relationship detection system is an adaptation of what is proposed in Girju [1]. The system uses a three step process in extracting treatment relationships which involves using pre-defined lexico-syntactic patterns to identify and extract treatment relationships, manually tagging the list of returned relationships to build up a training corpus and, finally, learning the classification rules to determine a valid relationship using a statistical classifier on the training corpus. We use a different approach to detect and extract treatment and causal relations by implementing a modified version of the Espresso system [3]. This attempts to accomplish the same goals as the Girju method but with much less manual supervision or building of a training set. We draw a direct comparison with Chu’s system because it is also a relationship-based patent retrieval system and is built on the same Drug Patent Corpus comprising of 50,000 United States medical patent documents. Since our system requires substantially lesser manual supervision than Chu’s implementation [4], we achieve an improvement in overall efficiency and performance.

5.3 Experimental Setup

Our relationship extraction algorithm was implemented in the Java programming language using MySQL as its backend database management system. All the test corpora are stored on the goliath.cs.ucla.edu server. We either remotely connected into our test schemas using a java.sql.connection object and ssh tunneling or directly connected to the Goliath server to run the tests. The outputs from the system were written to local files (stored on the local hard drive) or the Eclipse IDE console. The programming and tests were conducted on a Java ™ 6 environment and a MySQL database engine version 5.1. The tests were conducted on a personal computer having a 2.0 GHz Intel® Core™ 2 Duo processor with 3GB RAM. The tests took approximately 5-6 hours of running time, owing to the large size of the corpus used.

5.4 User Tests

We tested our relationship extraction system with the help of several users, roughly 15, from diverse backgrounds. Around 5 of these users were undergraduate students from UCLA, 2 from a non-Computer Science background and the remaining were graduate students from the Computer Science department at UCLA. Each of the users ran close to 600 tests i.e. they tested each of the total potential relations generated by the system and each of the relations retrieved by the system as above the threshold. This gave us nearly 3000 user tests to calculate the precision and recall for our system. For each relation, the users were asked to use their judgment to decide whether the relationship instance and the corresponding retrieved sentence for that treatment/causal verb was correct or not. These relations were the outputs generated by the system and written to local files.

5.5 Approach/Algorithm and Experiment – 1

5.5.1 Algorithm

In this test, the relationship extraction system was used to extract treatment relations from sentences in the Drug Patent Corpus. Using 16 input seed treatment relationship instances (see Appendix A for list of treatment input seeds); the algorithm recognized 15 verbs as being relevant to treatment. The verbs are determined to be relevant and are extracted by the system if they have an $r_{\prod}$ value greater than 0.2. The $r_{\prod}$ threshold of 0.2 was obtained by manually examining the results obtained after the verb extraction phase of the algorithm. The verbs extracted include synonyms and different degrees of the treatment verbs and the threshold is then applied on this entire set of verbs. The 15 top treatment verbs detected by this algorithm as being correct for the drug patent corpus are: lower, administer, inhibit, limit, block, relapse, decrease, suppress, lead, reduce, treat, result, acute, ameliorate and increase. These treatment verbs were then used to extract sentences from the corpus known to contain treatment relationships. Those sentences that contained one or more of these treatment verbs were chosen as candidates to perform extraction on.

5.5.2 Experiment

In this experiment, we chose 10 sentences (to maintain a manageable experiment set) for every relevant treatment verb for a total of 150 sentences, which in turn generated a total of 273 potential treatment relations. We calculated the relevancy score of each relationship instance and computed a threshold for these scores; the threshold was evaluated as the average relevancy score that resulted in a value of -0.3222 for this experiment. All the relations extracted were then manually assessed for correctness (including the original 273 relations). 136 relations were obtained with r_i score greater than the threshold out of which 88 were actually correct (as determined after manual tagging of all the instances). Of the original 273 relations, manual tagging determined that 140 of them were correct treatment relations. We finally calculated precision, recall and F-score of the system and the results obtained are displayed in Table 2 below:

Table 2. Results from Experiment 1

Precision	64.71%
Recall	62.86%
F-Score	63.77%

5.6 Approach/Algorithm and Experiment – 2

5.6.1 Algorithm

In this test, the relationship extraction system was used to extract both treatment and causal relations from sentences in the Drug Patent Corpus. The input seed relationship instances were extended to include both treatment and causal relationship instances giving a total of 24 input seeds (see Appendix B for list of treatment and causal input seeds). The algorithm recognized 29 verbs as being relevant to treatment and causal. The verbs are determined to be relevant and are extracted by the system if they have an $r_{\prod}$ value greater than 1.0. The $r_{\prod}$ threshold of 1.0 was obtained by manually examining the results obtained after the verb extraction phase of the algorithm. The verbs extracted include synonyms and different degrees of both the treatment and causal verbs and the threshold is then applied on this entire set of verbs. See Appendix C for a sample of the treatment and causal verbs extracted, their synonyms and the variation verbs added. The 29 top treatment and causal verbs detected by this algorithm as being correct for the drug patent corpus are: cure, alleviate, inhibit, limit, prevent, relieve, give, block, induce, contribute, outcome, decrease, suppress, provide, impact, reduction, consequence, related to, make, treat, result, therapeutic, ease, affect, stimulate, lead to, effect, remedy and increase. These treatment and causal verbs were then used to extract sentences from the corpus known to contain treatment and/or causal relationships. Those sentences that contained one or more of these verbs were chosen as candidates to perform extraction on.

5.6.2 Experiment

In this experiment, we again chose 10 sentences (to maintain a manageable experiment set) for every relevant treatment or causal verb for a total of 290 sentences, which in turn generated a total of 437 potential treatment relations. We calculated the relevancy score of each relationship instance and computed a threshold for these scores; the threshold was evaluated as the average relevancy score that resulted in a value of -0.7833 for this experiment. All the relations extracted were then manually assessed for correctness (including the original 437 relations). 207 relations were obtained with r_i score greater than the threshold out of which 134 were actually correct (as determined after manual tagging of all the instances). Of the original 437 relations, manual tagging determined that 169 of them were correct treatment relations. The results obtained from this experiment are displayed in Table 3 below:

Table 3. Results from Experiment 2

Precision	64.73%
Recall	79.29%
F-Score	71.27%

Table 4. Sample results of treatment verbs and retrieved sentences from Experiment 1

Treatment verb – inhibit **Retrieved relationship instance** - candidate compound ability, reaction **Retrieved sentence** - To test the ability of a candidate compound to **inhibit** binding, the reaction is run in the absence and in the presence of the test compound.
Treatment synonym verb - reduce (synonym to treatment verb limit) **Retrieved relationship instance** - formulation, reaction **Retrieved sentence** - Such formulations are said to **reduce** the adverse gastrointestinal reactions that may accompany oral tranexamic acid therapy (including nausea, vomiting, diarrhea, dyspepsia and cramping).

Table 5. Sample results of treatment and causal verbs and their retrieved sentences from Experiment 2

Treatment verb – cure **Retrieved relationship instance** - apparatus, emphysema **Retrieved sentence** - To the extent that the human lungs can rejuvenate under the conditions of no additional infections (as is the case for the inventor of William Banning Vail III, see the below), then the methods and apparatus necessary to remedy or partially **cure** emphysema are also disclosed in this invention.
Treatment synonym verb - ease (synonym to treatment verb relieve) **Retrieved relationship instance** - lignocamne, pain **Retrieved sentence** - Where necessary, the composition may also include a solubilizing agent and a local anesthetic such as lignocamne to **ease** pain at the site of the injection.
Causal synonym verb - contribute (synonym to causal verb give) **Retrieved relationship instance** - today, progression al **Retrieved sentence** - It is evident today that many of the factors which **contribute** to the progression of ALS are found in many other chronic and acute neurodegenerative disorders.
Causal variation verb – impact **Retrieved relationship instance** - modulators class, disease **Retrieved sentence** - Therefore, there is a potential for this class of modulators to **impact** angiogenesis-dependent diseases as well that may include among others, diabetic retinopathy, macular degeneration, obesity and inflammatory disease such as rheumatoid arthritis.

5.7 Results and Inferences

Tables 4 and 5 indicate samples of the verbs extracted, a corresponding relationship instance retrieved by the system and the sentence in which the instance occurred. From the results of Experiment 2, we observe that there is more than a 15% of considerable increase in the recall from Experiment 1. This indicates that when causal relations are added along with treatment relations in the input seeds, the number of treatment and causal verbs extracted increases; thereby translating to an increase in the number of relationship instances and the number of relations extracted above the threshold value. Although some incorrect relations are extracted, the overall improvement in the recall of the system makes up for the slightly low precision.

Figure 1 gives us a clear picture of the differences in the recall values for Chu's classification rules, his patent retrieval system [4], and our extraction system with the treatment relationships alone and with both treatment and causal relationships. Comparing with Chu's overall system [4], we see that the classification rules produce a high precision of 81.94% but a very low recall of only 47.97%. One of the reasons for the low recall of classification rules stated by the authors of [4] is that the presence of highly technical medical terms in the corpus makes it difficult to find hierarchical noun classes for them. Our system fares better in this aspect since we use the UMLs medical dictionary for this purpose. However, Chu's patent retrieval system, on the other hand, evaluates to a precision of 85.81% and a recall of 82.71%.

The authors explain the exceptionally high recall with two reasons. Firstly, the keyword terms used in the test queries are constrained to actual treatment due to which the relationships extracted naturally conform to treatment relationships. Secondly, the recall projected to be the system's recall is not the global recall. This is due to the fact that the treatment relationships extracted are confined to those of the structure NP_1 VP NP_2. Both these factors have been addressed in our relationship extraction system, which uses only input seeds of a particular relationship type to extract a certain kind of verbs and relationship instances. It also addresses 91.2% of all binary relations as claimed by Banko [2] instead of restricting to relationships of the form NP_1 VP NP_2 only.

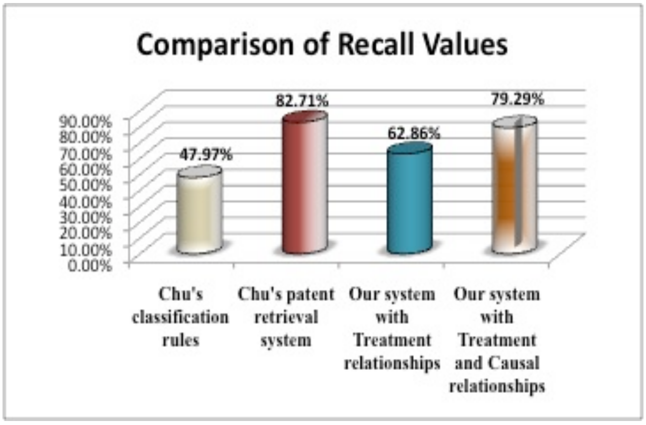

Figure 1. Comparison of Recall values between Chu's system and Experiments 1 and 2

The Espresso method is almost completely automated and requires minimal human supervision in order to extract and output relations from text. The only input required to the algorithm is a small set of seed relations representing known correct relations of a particular relationship type. As a result, it may extract some incorrect relation instances to be relevant, causing a drop in precision. On the other hand, in the Girju method used by Chu's system [4], verbs extracted by the system are manually examined for correctness and only key verbs are retained as correct treatment verbs. Furthermore, thousands of extracted relationship instances have to be manually tagged and then fed into a statistical classifier in batches to determine the final classification rules. Thus, all the extra manual effort that goes into extracting relationships explains the reason for Chu's system having a higher precision score. While our system loses some points in precision, the complete automation and minimal human effort along with high recall make up for the loss of precision.

6. CONCLUSIONS

In this paper, we examine how the recall of a patent retrieval system can be improved by using a relationship extraction based search instead of a keyword-based search. Treatment relationships are those, which describe instances where A can be used to treat B, whereas causal relationships are those, which describe instances where C causes B. However, we are unable to make a one-on-one comparison with a keyword-based retrieval system, as no common patent corpus was available to us that had been used in both types of systems.

Firstly, we extract treatment relationships alone from the corpus and use these to retrieve the patent documents. In our second experiment, we use both treatment and causal relationships as a basis to retrieve the relevant patent documents. In our extraction system, we increase the set of binary relationships addressed to include 91.2% of the relationships that can occur in a document. From the experiments, we also notice that there is a significant increase in the recall of the system when we include synonyms and variations of the extracted verbs. We attribute this increase to the fact that the expansion in the verbs increases the search space thereby pulling out more relationship instances from the corpus.

Ideally, we would like to have a system with high precision and recall. However, we believe that if we have to choose between the two, it is more important in the case of a patent retrieval system to have higher recall than higher precision. This is because we do not want to lose relationship information present in a patent document even if that means we get relationship results that may not always be correct. In other words, it is better to get most of the relationships from a document (including a few that are incorrect) than to not recognize a bulk of the correct relationships present in the patent document. Specifically, when patent researchers need to find out prior-art patents or when patent applicants want to find out contending technologies of other sources, a higher recall but lower precision system will be acceptable because although it may retrieve some inaccurate patents this is better than leaving out patents which are relevant – if an ideal system with top recall and top precision is not possible.

7. REFERENCES

[1] Girju, R. 2003. Automatic Detection of Causal Relations for Question Answering. *In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003), Workshop on "Multilingual Summarization and Question Answering - Machine Learning and Beyond".* 2003.

[2] Banko, M. and Etzioni, O. 2008. The Tradeoffs Between Open and Traditional Relation Extraction. *In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008).*

[3] Pantel, P. and Pennacchiotti, M. 2006. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. *In Proceedings of Conference on Computational Linguistics / Association for Computational Linguistics* (COLING/ACL - 06), pp. 113-120. Sydney, Australia.

[4] Chu, A., Sakurai, S. and Cárdenas, A. F. 2008. Automatic detection of treatment relationships for patent retrieval. *In Proceeding of the 1st ACM workshop on Patent information retrieval*, October 30, Napa Valley, California, USA.

[5] Ravichandran, D. and Hovy, E. H. 2002. Learning surface text patterns for a question answering system. *In Proceedings of ACL-2002, pp. 41-47*. Philadelphia, PA.

[6] A. Yates and O. Etzioni. 2007. Unsupervised resolution of objects and relations on the web. *In Procs of NAACL/HLT.*

[7] The Penn Treebank project. http://www.cis.upenn.edu/~treebank/.

[8] Larkey, L. 1999. A patent search and classification system. International Conference on Digital Libraries archive, *In Proceedings of the fourth ACM conference on Digital libraries table of contents*, Berkeley, California, United States, Pages: 179-187.

[9] Larkey, L., Connell, M. and Callan, J. 2000. Collection selection and results merging with topically organized U.S. patents and TREC data. *In Proceedings of the 9th International Conference on Information and Knowledge Management (CIKM), ACM*, Pages: 282-289.

[10] Wu, T., Holzman, L. E., Pottenger, W. M., Phelps, D. J. 2003. A Supervised Learning Algorithm for Information Extraction from Textual Data. *In Proceedings of the workshop on Text Mining, Third SIAM International Conference on Data Mining*.

[11] Wanner, L., Baeza-Yates, R., Brügmann, S., Codina, J., Diallo, B., Escorsa, E., Giereth, M., Kompatsiaris, Y., Papadopoulos, S., Pianta, E., Piella, G., Puhlmann, I., Rao, G., Rotard, M., Schoester, P., Serafini, L., Zervaki, V. 2008. Towards Content-Oriented Patent Document Processing. *World Patent Information*, Elsevier, Vol. 30(1), Page: 21-33.

[12] Tiwari, A. and Bansal, V. 2004. PATSEEK: Content Based Image Retrieval System for Patent Database. *In Proceedings of the International Conference on Electronic Business ICEB*, pages 1167–1171.

[13] Vrochidis, S., Papadopoulos, S., Moumtzidou, A., Sidiropoulos, P., Pianta, E. and Kompatsiaris, I. 2010. Towards Content-based Patent Image Retrieval; A Framework Perspective. *World Patent Information Journal*, Volume 32, Issue 2, pp 94-106.

Appendix A
Input Seeds used to extract Treatment Relationships

(Xanax, Anxiety)
(Ambien, Insomnia)
(Effexor, Depression)
(Paxil, Depression)
(Lexapro, Depression)
(Caffeine, Depression)
(Zoloft, Depression)
(Imipramine, Depression)
(Glycoside, Depression)
(Ibuprofen, Arthritis)
(Ibuprofen, Headache)
(Tylenol, Fever)
(Tylenol, Headache)
(Antibody, Inflammation)
(Ibuprofen, Inflammation)
(Surgery, Glaucoma)

Appendix B
Input Seeds used to extract Treatment and Causal Relationships

(Xanax, Anxiety)
(Ambien, Insomnia)
(Effexor, Depression)
(Paxil, Depression)
(Lexapro, Depression)
(Caffeine, Depression)
(Zoloft, Depression)
(Imipramine, Depression)
(Glycoside, Depression)
(Ibuprofen, Arthritis)
(Ibuprofen, Headache)
(Tylenol, Fever)
(Tylenol, Headache)
(Antibody, Inflammation)
(Ibuprofen, Inflammation)
(Surgery, Glaucoma)
(Cardiac arrest, Heart attack)
(Diabetes, Sugar)
(Anxiety, Genes)
(Anxiety, Chemical imbalance)
(Insomnia, Anxiety)
(Insomnia, Stress)
(Insomnia, Depression)
(Insomnia, Hormonal change)

Appendix C
Sample of full list of verbs used in Experiment 2

Treatment Verbs and their Synonyms

Verb: cure
Synonym: curative
Synonym: remedy
Synonym: therapeutic
Synonym: bring around
Synonym: heal

Verb: inhibit
Synonym: conquer
Synonym: curb
Synonym: stamp down
Synonym: subdue
Synonym: suppress
Synonym: bottle up

Verb: limit
Synonym: bound
Synonym: confine
Synonym: restrain
Synonym: restrict
Synonym: throttle
Synonym: fix

Verb: prevent
Synonym: forbid
Synonym: foreclose
Synonym: forestall
Synonym: preclude

Verb: relieve
Synonym: alleviate
Synonym: assuage
Synonym: palliate
Synonym: allay
Synonym: ease
Synonym: lighten

Verb: block
Synonym: bar
Synonym: barricade
Synonym: stop
Synonym: hinder
Synonym: obstruct
Synonym: stymie
Synonym: halt
Synonym: jam
Synonym: impede
Synonym: occlude
Synonym: blank out
Synonym: immobilise
Synonym: immobilize

Verb: decrease
Synonym: drop-off
Synonym: lessening
Synonym: decrement
Synonym: reduction
Synonym: diminish
Synonym: fall
Synonym: lessen
Synonym: minify

Verb: suppress
Synonym: conquer
Synonym: curb
Synonym: inhibit
Synonym: stamp down
Synonym: subdue
Synonym: crush
Synonym: oppress
Synonym: bottle up
Synonym: repress

Verb: treat
Synonym: care for

Causal Verbs and their Synonyms

Verb: give
Synonym: yield
Synonym: render
Synonym: feed
Synonym: contribute

Verb: make
Synonym: induce
Synonym: stimulate
Synonym: cause
Synonym: create
Synonym: bring in
Synonym: get
Synonym: have

Verb: result
Synonym: consequence
Synonym: effect
Synonym: outcome
Synonym: ensue

Verb: affect
Synonym: bear upon
Synonym: impact
Synonym: involve

Verb: increase
Synonym: gain
Synonym: increment

Causal Variation Verbs added

Verb: lead to
Verb: related to
Verb: positively impact