NICTA and UBC at the TREC 2012 Medical Track

David Martinez[†] Arantxa Otegi* Eneko Agirre*
[†]NICTA and CIS Department, University of Melbourne, Australia
*IXA Group, University of the Basque Country UPV/EHU, Donostia, Basque Country

Abstract

We introduce two heterogeneous query expansion techniques, and a combined system to the TREC 2012 Medical Track. Our methods are based on external resources that provide expansion concepts related to the query terms, by means of the PageRank algorithm, and simple rules based on UMLS Semantic Types. In this paper we show that our systems are able to reach competitive performances at both the TREC-2011 and TREC-2012 tasks.

1 Introduction

In this paper we present the combined submission of the teams NICTA and UBC, which focuses on query expansion techniques. For this edition we build upon the NICTA-2011 systems [8], and we incorporate the Personalised PageRank algorithm in order to select the most similar concepts to the query terms, and then use them for query expansion.

The NICTA system was ranked 6th on the 2011 Medical Track (with regards to the Bpref measure). We did minimal changes to this knowledge-based query expansion method, and centered our efforts in combining this technique with a graph-based expansion approach from the UBC team, namely Personalised PageRank.

Personalized PageRank [6] has been successfully used in Natural Language Processing tasks such as Word Sense Disambiguation [3, 4, 5, 11] and word similarity [1, 2]. It has been applied both to a general purpose lexical knowledge-base such as WordNet [1, 2, 3, 4] and also to UMLS [5, 11]. In addition, recent results show that it is useful to improve ad-hoc IR with WordNet [9]. In this work, we apply it on UMLS in order to improve results over the TREC task.

Our final scores show that query expansion is beneficial over the baseline methods; specially over the TREC-2011 queries, where it reaches the performance of the best 2011 systems. For the TREC-2012 query-set the results were far from the best performing system, but above the median of the submissions.

2 Method

We present here the steps of our approach: we start by describing how we processed the TREC document collection; then we explain our query processing method, including the expansion techniques; finally we detail our indexing and searching approaches.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
NICTA and UBC at the TREC 2012 Medical Track	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Melbourne,NICTA and CIS Department,Australia,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the Twenty-First Text REtrieval Conference (TREC 2012) held in Gaithersburg, Maryland, November 6-9, 2012. The conference was co-sponsored by the National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA). U.S. Government or Federal Rights License

14. ABSTRACT
We introduce two heterogeneous query expansion techniques, and a combined system to the TREC 2012 Medical Track. Our methods are based on external resources that provide expansion concepts related to the query terms, by means of the PageRank algorithm, and simple rules based on UMLS Semantic Types. In this paper we show that our systems are able to reach competitive performances at both the TREC-2011 and TREC-2012 tasks.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**12**	

Field	Description
ADMITDIAG	diagnostics during admission
AGE	patients age by decades, for example age30 means people in their thirties
ALLERGIES	allergies listed in the report
CHIEFCOMP	chief complaint, this may be equal to diagnostics during admission
DISCHDIAG	discharge diagnostics
GENDER	patient's gender extracted from text and represented as gendermale and genderfemale
HISTORY	history of the patient's medical condition or past medical illness
MEDICATIONS	medications
PRESTHIS	present illness medical history
PASTHIS	past medical history
REPORT	all the free text information, including history, past and present, and allergies

Table 1: List of fields defined for Boolean search.

2.1 Processing the Document Collection

We apply the same pipeline as in [8] for processing the document collection. We start by expanding the mentions of ICD9 codes[1] of admission and discharge diagnoses in the metadata with their text descriptions. Both the original code and expanded forms were included for indexing.

The documents contain different sections, with their corresponding headings. We rely on hand-crafted pattern-matching rules to identify the main headings, in order to build different indices and allow for field-based search. The list of the fields we cover is given in Table 1. Apart from these fields, we built rules to identify and normalise some demographic information, such as gender, age, and other specific conditions (such as weight) mentioned in the text.

We also ran NegEx[2] over the entire collection in order to detect negated phrases. We rely on the in-built NegEx parser of *MetaMap-2010*, which specifies which of the identified phrases appear to be negated. We use this information to build an index that converts negated terms that are majority in a given document, into a new representation, where the negated phrase is transformed into a single word, with no space, and with a "*no*" prefix: e.g., if negation is implied for "chronic back pain", all instances of "chronic back pain" are replaced with the word "nochronicbackpain". Our aim with this index is to avoid matching cases where the term appears negated in the document more often than as positive. Due to the lack of negated queries in the collections, the result of transforming words is the same as removing them.

Finally, we indexed the collection with and without the Porter stemmer.

2.2 Processing Queries

We describe first our methods to identify fields in the query, and then our different expansion approaches.

2.2.1 Identifying Fields in the Query

We developed a set of manually constructed patterns to map query terms into the available fields (Table 1). These patterns — formed based on the sample clinical questions provided by the National Library of Medicine (NLM) [7] — covered seven broad categories of age,

[1] International Statistical Classification of Diseases and Related Health Problems: `http://en.wikipedia.org/wiki/List_of_ICD-9_codes`

[2] `http://code.google.com/p/negex/`

What	Pattern	Translation
Gender	women/female	GENDER:genderfemale
	men/male	GENDER:gendermale
Age	young adult	AGE:(age20 age30 age40)
	younger/young	AGE:(agebirth12 ageteen age20 age30 age40)
	adult	AGE:(age20 age30 age40 age50 age60 age70 age80 age90)
Weight	(BMI|Body Mass Index) (bigger than|more than|of|approximately of)	
	>= 36	WEIGHT:(obesity obese overweight "morbidly obese" "morbid obese" "morbid obesity" "markedly obese")
	>=30 and <=35	WEIGHT:(obesity obese overweight "moderately obese" "moderate obesity")
	>=25 and <=30	WEIGHT:(obesity obese overweight "slightly obese" "mildly obese")
	>=18.5 and <=25	WEIGHT:("normal weight")
	(BMI|Body Mass Index) (less than|of|approximately of)	
	>16 and <=18.5	WEIGHT:(underweight)
	<=16	WEIGHT:(underweight "severely underweight")
Treatments	taking X (who|with|without|treated)	MEDICATIONS:X
	who are on X	MEDICATIONS:X
	patients on X for Y	MEDICATIONS:X
Admission Diagnostics	admitted (for|with) X who	CHIEFCOMP:X OR ADMITDIAG:X
	treated for X (who|during|while)	PRESTHIS:X OR DISCHDIAG:X
	(patients with|men with|women with) X	PRESTHIS:X OR DISCHDIAG:X
	who were discharged X	DISCHDIAG:X
History	with a* history of X (who|now)	HISTORY:X
Allergy	with X allergy	ALLERGY:X
	without allergy	ALLERGY:(noallergies)
Abbreviation	seen in the er|presented to the er	REPORT:("emergency room" OR ER)

Table 2: Rules (patterns in the queries and their translations) used in the query transformation step. Words that are all in capital letters are field names.

weight (using body mass index), diagnostics, treatments, medications, history, allergies, and abbreviations. For example, if a query contained "elderly patients", we expanded "elderly" with an equivalent age field that covered people in their 60s to 90+. Table 2 shows the details of the selected transformation rules. For example the query:

Elderly patients with ventilator-associated pneumonia

is translated to:

> *PRESTHIS:(ventilator associated pneumonia) OR DISCHDIAG:(ventilator associated pneumonia) OR AGE:(age60 age70 age80 age90) OR REPORT:(elderly with ventilator associated pneumonia).*

A small number of abbreviations, such as ER (emergency room), were also expanded in the queries.

2.2.2 Query Expansion using Semantic Types (ST)

We leveraged external resources to add new terms to our queries, by identifying terms that are strongly related to the query terms. Specifically, we focused on query terms that represent

medical categorical concepts (e.g. disease categories). For example, for the query below, we added terms falling under the category of "atypical antipsychotics":

Patients taking atypical antipsychotics

Our approach to expansion used two main knowledge sources: the UMLS Metathesaurus (version 2010AA) and DBpedia. In order to select expansion candidates we used *MetaMap-2010* from the National Library of Medicine (NLM). We defined manual expansion rules from these resources based on the sample queries of TREC-2011 and 50 queries from the priority list from the US Institute of Medicine of the National Academies[3].

Using these queries, we defined a small set of stop-categories that would have otherwise produced undesirable expansions. The following terms were excluded from expansion: "administration", "AMA", "diagnosis", "drug", "functional concept", "medication", and "surgery". We also removed terms with the following strings from the DBpedia output: "code", "history", "mechanism", "poisoning", "toxicity", and "withdrawal".

During the development process, we explored expansion using hierarchical relations from the UMLS Metathesaurus, by selecting all the terms in the hyponym concepts; however, we observed that DBpedia offered a higher coverage of some domains, such as newly developed drugs, and it also showed less risk of over-expansion. For instance, one sample query contained the term "atypical antipshychotic", which UMLS expanded with 8 more specific drugs (e.g. "Clozapine"). DBpedia, however, identified the same set of drugs, as well as a further 22 new drug and brand names, which seemed correct after manual analysis, and had a stronger presence in the collection.

For our final expansion system, we first applied MetaMap to identify phrases linked to terms in the UMLS Metathesaurus. The matched concepts were then used as candidate terms to be expanded; in some cases terms consisted of a primary term followed by a parenthesized description — such as "Intervention (Surgical and medical procedures)" — and in such cases we treated them as separate candidate terms.

Each candidate term had a Semantic Type (ST) associated with it in the MetaMap output. We used STs to define two expansion groups: safe expansion (for terms which STs include the string "Pharmacologic Substance") and filtered expansion (for terms whose ST is "Therapeutic or Preventive Procedure"). Candidate terms that did not belong to these groups were discarded; for the rest, if they were listed as "category" in DBpedia[4], we extracted all of the terms listed under the category as our expansion terms. Then, for "safe expansion" the output was the full list of expansion terms; for "filtered expansion", we removed terms which are not UMLS concepts by applying MetaMap to each term.

2.2.3 Query Expansion using Personalised PageRank

For this approach, we use a graph algorithm based on random walks over the graph representation of a knowledge-base of concepts and relations, to obtain concepts related to the queries. The UMLS Metathesaurus is used as the knowledge-base, and we represent UMLS as a graph.

Apart from concepts, UMLS Metathesaurus also contains a wide range of information about the relations between concepts in the form of database tables. The MRREL table

[3] http://www.iom.edu/~/media/Files/Report\%20Files/2009/ComparativeEffectivenessResearchPriorities/Stand\%20Alone\%20List\%20of\%20100\%20CER\%20Priorities\%20-\%20for\%20web.pdf

[4] http://wiki.dbpedia.org/OnlineAccess

lists relations between concepts like "parent", "can be qualified by" or "related and possibly synonymous" among others. The MRCOC table contains co-occurrence relations between concepts, that is, relations between similar concepts or different concepts that share an important connection. In order to obtain the graph structure of UMLS, we simply treat the concepts in UMLS as vertices, and the relations listed in the MRREL and MRCOC tables as edges. No weights are used for the relations that are extracted from the MRREL table.

Given a query and the graph-based representation of UMLS, we obtain a ranked list of related concepts as follows:

1. We first run MetaMap and identify the UMLS concepts in the query, we explore two variants: with and without the in-built Word Sense Disambiguation (WSD) module. We also rely on the NegEx module to remove negated concepts. Note that in cases where we rely on field search, we treat each field as a separate query for this kind of expansion.

2. We then assign a uniform probability distribution to the concepts found in the query. The rest of nodes are initialized to zero.

3. We compute personalized PageRank [6] over the graph, using the previous distribution as the initial distribution, and we produce a probability distribution over UMLS concepts. The higher the probability for a concept, the more related it is to the given text.

Basically, personalized PageRank is computed by modifying the random jump distribution vector in the traditional PageRank equation. In our case, we concentrate all probability mass in the concepts identified in the query.

Let G be a graph with N vertices $v_1, \ldots, v_N$ and d_i be the outdegree of node i; let M be a $N \times N$ transition probability matrix, where $M_{ji} = \frac{1}{d_i}$ if a link from i to j exists, and zero otherwise. Then, the calculation of the *PageRank vector* $\mathbf{Pr}$ over G is equivalent to resolving Equation (1).

$$\mathbf{Pr} = cM\mathbf{Pr} + (1 - c)\mathbf{v} \tag{1}$$

In the equation, $\mathbf{v}$ is a $N \times 1$ vector and c is the so-called *damping factor*, a scalar value between 0 and 1. The first term of the sum on the equation models the voting scheme described in the beginning of the section. The second term represents, loosely speaking, the probability of a surfer randomly jumping to any node, e.g. without following any paths on the graph. The damping factor, usually set in the [0.85..0.95] range, models the way in which these two terms are combined at each step.

The second term on Eq. (1) can also be seen as a smoothing factor that makes any graph fulfill the property of being aperiodic and irreducible, and thus guarantees that the PageRank calculation converges to a unique stationary distribution.

In the traditional PageRank formulation the vector $\mathbf{v}$ is a stochastic normalized vector whose element values are all $\frac{1}{N}$, thus assigning equal probabilities to all nodes in the graph in case of random jumps. In the case of personalized PageRank as used here, $\mathbf{v}$ is initialized with uniform probabilities for the concepts in the query, and 0 for the rest of terms.

PageRank is actually calculated by applying an iterative algorithm which computes Eq. (1) successively until a fixed number of iterations are executed. In our case, we used a publicly

available implementation[5], with the default values provided by the software, i.e. a damping value of 0.95, and 30 iterations.

In order to select the expansion terms from the ranking of concepts, we use a threshold value to retrieve the top concepts, and then we obtain all the terms that appear under each concept in the UMLS Metathesaurus. We explored two approaches to determine the cut-off value: (i) select the top k concepts, or (ii) select all the concepts with weights above a given t threshold. Our preliminary experiments over the TREC-2011 dataset suggested that the former approach was able to provide better performances for different settings, and we decided to use the top k concepts for our experiments.

2.2.4 Combined Query Expansion

In order to combine our two different expansion techniques, we can simply merge the terms from each expansion source into a joint query. Another approach that we explored is to rely on the expanded terms from the ST-expansion to initialise the PageRank method. We report the results of the two methods in our experiments.

2.3 Indexing and Searching

We first distinguish between two types of indexing in our runs: *visit-based* and *report-based*. In the former approach, all related reports for a visit were concatenated (removing duplicate diagnostics codes) to create a single "multi-document" item for indexing. We refer to the former approach as VISIT, and as REPORT to the latter.

As explained in Section 2.1, we also generate different indexes depending on the use of separate fields or not (FIELDS/COMBINED), or the application of stemming (STEM/NOSTEM). When we rely on field search, a Boolean search over the fields is followed by ranking.

We used stop-word removal both in query processing and indexing; however, we augmented the typical list of stop-words with *patient*, and removed single characters, *and*, *or*, *not*, and *no* from the list.

Regarding negation, as explained in Section 2.1, we pre-processed the document collection with NegEx, in order to to handle negated terms, and built separate indices. Since we only observed minor differences, we settled on a single index for each of the collections. Thus, we report here the results using the NegEx-processed index for TREC-2011, and the full index for TREC-2012.

The search engine used for indexing and searching in our runs was Apache Lucene (v3.2); we used both the BM25 and *tf-idf* ranking algorithms for Lucene [10].

3 Results over the TREC-2011 query set

We first tested different combinations of our main approaches over the TREC-2011 query set and collection, in order to select the most promising configurations for TREC-2012. We relied on the same evaluation metric that was used in TREC-2011: Bpref.

We performed three main experiments:

- PageRank without Semantic Type (ST) expansion

[5] `http://ixa2.si.ehu.es/ukb/`

System	WSD	Best Bpref (threshold)	Worst Bpref (threshold)
Base system	No	0.5218	**0.5218**
PageRank first	No	**0.5438** (3)	0.5203 (18)
PageRank first	Yes	0.5373 (9)	0.5026 (3)
Query Transformation first	No	0.5427(15)	0.3719 (3)
Query Transformation first	Yes	0.5412 (8)	0.4048 (3)

Table 3: Performance of different PageRank settings over the TREC-2011 query set (VISIT+STEM+COMBINED and TF-IDF), together with the corresponding baseline (no expansion). Best results per column in bold.

System	WSD	Best Bpref (threshold)	Worst Bpref (threshold)
Base system	No	0.4973	**0.4973**
PageRank first	No	**0.5601** (3)	0.4959 (19)
PageRank first	Yes	0.5162 (8)	0.4771 (15)
Query Transformation first	No	0.5539 (7)	0.4023 (3)
Query Transformation first	Yes	0.5270 (8)	0.4218 (3)

Table 4: Performance of different PageRank settings over the TREC-2011 query set (REPORT+NOSTEM+COMBINED and TF-IDF), together with the corresponding baseline (no expansion). Best results per column in bold.

- Combine ST expansion and PageRank, without field indexing
- Combine ST expansion and PageRank, with field indexing

As mentioned above, when we combine PageRank and ST, we have to choose if we want to apply PageRank over the query concepts, or over the ST-expanded concept set. We present the results for the two different settings in most of our experiments. There are other two alternatives when applying PageRank: to perform WSD prior to choosing the initial concepts, or not to use WSD. We report here the results of the two variants. Finally, we also need to set a threshold to decide the number of top concepts to use. As mentioned above, we performed preliminary experiments using two types of thresholds: weight-based (i.e. choose all the concepts above the cut-off PageRank weight) and ranking-based (i.e. select all the concepts in the top k positions), and settled on the latter setting. We report the results for the best and worst cut-offs in the range 3-20 over the TREC-2011 dataset.

We start our analysis by evaluating the performance of PageRank without ST expansions. In this case we also need to decide whether we parse the query before applying PageRank or not. For our first experiment we chose the index VISIT+STEM+COMBINED and TF-IDF ranking as basic system, since it achieved the highest performance in previous experiments when no ST expansions were used.

The results over the TREC-2011 query set are given in Table 3, together with the basic system without PageRank. We can see that the system achieves its best performances when applying PageRank first, and that we are able to improve over the baseline. WSD does not seem to be helpful, and starting with all the concepts from MetaMap (not only the disambiguated ones) is the best strategy for this experiment.

System	WSD	Best Bpref (threshold)	Worst Bpref (threshold)
Base system	No	0.5218	0.5218
ST Expansion	No	0.5078	0.5078
PageRank first	No	**0.5655** (3)	**0.5293** (18)
PageRank first	Yes	0.5488 (3)	0.5277 (20)
ST Expansion first	No	0.5501(9)	0.4923 (3)
ST Expansion first	Yes	0.5480 (5)	0.4997 (3)

Table 5: Performance of different combinations of PageRank and ST expansions over the TREC-2011 query set (VISIT+STEM+COMBINED and TF-IDF), together with the corresponding baselines (no expansion and ST expansion). Best results per column in bold.

System	WSD	Best Bpref (threshold)	Worst Bpref (threshold)
Base system	No	0.4973	0.4973
ST Expansion	No	0.4895	0.4895
PageRank first	No	**0.5789** (3)	**0.5422** (10)
PageRank first	Yes	0.5495 (3)	0.5226 (10)
ST Expansion first	No	0.5642 (7)	0.5008 (4)
ST Expansion first	Yes	0.5468 (5)	0.5041 (3)

Table 6: Performance of different combinations of PageRank and ST expansions over the TREC-2011 query set (REPORT+NOSTEM+COMBINED and TF-IDF), together with the corresponding baselines (no expansion and ST expansion). Best results per column in bold.

Next we performed a similar experiment by using report indexing (instead of visits), and no stemming; we chose this indexing because it was also competitive, and we observed clear differences over the outputs of these settings in previous experiments. We present the results of this experiment in Table 4. We can see that the results are similar to the previous experiment, and we also observe an increase in the best Bpref value.

For our next experiment we combine the ST expansion with PageRank. As base configuration, we rely on the same index and ranking used in the previous test (VISIT+STEM+COMBINED and TF-IDF). The results of this experiment are given in Table 5. There is a larger improvement over the baseline in this case, and even the worst thresholds improve the baseline when PageRank is applied first. Note that the best results are similar to the best official submission for the TREC 2011 challenge. Again, the best performance is achieved without WSD.

We also apply the alternative baseline system of report indexing and no stemming for the combined system. We present the results of this experiment in Table 6. These results reach the highest Bpref score so far, and are more robust regarding the lower bounds. Again, the best strategy is to apply PageRank first, and not to use WSD in the process.

We then explore the use of fields in the indexing. This approach obtained worse performance that combining fields in our previous experiments, and we only perform two runs, always applying PageRank first. The results are shown in Table 7. We can see that the gains are smaller than in previous configurations, and there is a big drop in the case of the worst threshold.

System	WSD	Best Bpref (threshold)	Worst Bpref (threshold)
ST Expansion	No	0.4802	**0.4802**
PageRank first	No	**0.5127** (7)	0.4561 (19)
PageRank first	Yes	0.4955 (7)	0.4540 (19)

Table 7: Performance combining PageRank and ST expansions over the TREC-2011 query set using FIELDS, together with the ST expansion baseline. Best results per column in bold.

Expansion	Configuration	Best Bpref (thr.)	Worst Bpref (thr.)
No expansion	VISIT+STEM+COMBINED and TF-IDF	0.5218 (-)	0.5218 (-)
	REPORT+NOSTEM+COMBINED and TF-IDF	0.4973 (-)	0.4973 (-)
ST	VISIT+STEM+COMBINED and TF-IDF	0.5078 (-)	0.5078 (-)
	REPORT+NOSTEM+COMBINED and TF-IDF	0.4895 (-)	0.4895 (-)
PageRank	VISIT+STEM+COMBINED and TF-IDF	0.5438 (3)	0.5203 (18)
	REPORT+NOSTEM+COMBINED and TF-IDF	0.5601 (3)	0.4959 (19)
Combined	VISIT+STEM+COMBINED and TF-IDF	0.5655 (3)	0.5293 (18)
	REPORT+NOSTEM+COMBINED and TF-IDF	**0.5789** (3)	**0.5422** (10)

Table 8: Summary of results over the TREC-2011 query set for the types of systems we developed. thr: threshold when available.

We summarise the best results of our systems over the TREC-2011 dataset in Table 8. We observe that PageRank expansion helps to improve the baseline, and that the best performance is obtained when combining it with ST, even if ST alone does not perform well.

4 Official results over the TREC-2012 query set

At the time of submitting the runs, we did not have all the information regarding the optimal values of combinations and parameters, so we chose four configurations that had achieved good performance over the TREC-2011 dataset at the time. For all our runs, we relied on the COMBINED index (since FIELDS did not perform well over TREC-2011), and we did not process negations for the documents (only for the queries), we also use TF-IDF in all the submitted runs:

- NICTAUBC1: Combined expansion, PageRank first (threshold = 3), index REPORT+STEM
- NICTAUBC2: Combined expansion, ST expansion first (threshold = 4), index REPORT+NOSTEM
- NICTAUBC4: ST expansion, index VISIT+STEM
- NICTAUBC6: Combined expansion, ST expansion first (threshold = 6), index VISIT+NOSTEM

The results of the different systems are given in Table 9, together with the median and best results for the automatic runs. We can see that NICTAUBC4 is our best performing

System	infAP	infNDCG	Bpref	Position
NICTAUBC1	0.1947	0.4362	0.3455	27
NICTAUBC2	0.1912	0.4450	0.3457	26
NICTAUBC4	**0.2162**	**0.4870**	**0.3771**	11
NICTAUBC6	0.1837	0.4193	0.3380	33
Best Automatic	0.4238	0.7461	0.4515	1
Median Automatic	0.1695	0.4243	0.3288	41

Table 9: Official results for TREC-2012. Position indicates the ranking of the system among the 82 automatic runs. Best results in bold. Note that these runs were submitted before completing the experiments in the previous section, and are not optimal.

Expansion	Configuration	infAP	infNDCG	Bpref	Position
No expansion	VISIT+STEM+COMBINED and TF-IDF	0.1744	0.3860	0.3205	44
PageRank	VISIT+STEM+COMBINED and TF-IDF (thr=3)	0.1994	0.4340	0.3542	20
ST	VISIT+STEM+COMBINED and TF-IDF	**0.2162**	**0.4870**	**0.3771**	11
Combined	REPORT+NOSTEM+COMBINED and TF-IDF (thr=3)	0.1901	0.4383	0.3455	27

Table 10: Results over TREC-2012 using the best parametrization for each technique, as attested in TREC-2011.

system, scoring well above the median for all metrics. This means that we obtained the best performance when we relying on ST expansion alone, and unlike our TREC-2011 experiments, combining ST and PageRank did not help.

5 Additional experiments

After the qrels were released, we already had obtained the complete set of results on TREC 2011, and we performed additional experiments.

We first checked the performance of the best 2011 configurations (cf. Table 8) on the 2012 data. Table 10 shows those results, with both PageRank and ST improving over the baseline system, confirming that they are successful strategies for expansion. For TREC-2012, ST seems to be the best strategy, and surprisingly, the combination of ST and PageRank does not perform so well.

Alternatively, we also wanted to check the results of the algorithms when the parameters and combinations are optimized using TREC-2012 as development, and TREC-2011 as the test dataset. Table 11 shows the the optimal configurations and performances for each of the types of expansions that we tested over the TREC-2012 dataset. We observe that PageRank is able to match ST's performance, given the optimal setting, and the best results are obtained combining both, except for infNCDG, which reports the best results for ST.

Table 12 reports the results on the TREC-2011 dataset, when the parameters are those obtained in the optimisation over TREC-2012. Note that Bpref is the only measure available on the 2011 data. The results confirm that both expansion strategies (PageRank and ST) are useful, with the best results obtained with PageRank, and the combination yielding the best results.

Expansion	Configuration	infAP	infNDCG	Bpref
No expansion	REPORT+STEM+FIELDS and BM25	0.1793	0.4168	0.3381
PageRank	REPORT+STEM+COMBINED (thr=4) and TF-IDF	0.2176	0.4704	0.3771
ST	VISIT+STEM+COMBINED and TF-IDF	0.2162	**0.4870**	0.3771
Combined	VISIT+STEM+COMBINED (thr=4) and TF-IDF	**0.2252**	0.4790	**0.3880**

Table 11: Upperbound of results over TREC-2012 as obtained when using the optimal parameters for each method.

Expansion	Configuration	Bpref
No expansion	REPORT+STEM+FIELDS and BM25	0.4160
PageRank	REPORT+STEM+COMBINED (thr=4) and TF-IDF	0.5469
ST	VISIT+STEM+COMBINED and TF-IDF	0.5078
Combined	VISIT+STEM+COMBINED (thr=4) and TF-IDF	**0.5521**

Table 12: Results over TREC-2011 using the best parametrization for each technique, as attested in TREC-2012.

6 Conclusions

This year We have tested two different methods for query expansion based on DbPedia and UMLS. The first method is heuristic query expansion, and the second is based on random walks over UMLS. Our development experiments on TREC-2011 showed that our heuristic and random-walk expansion algorithms (ST and PageRank, respectively) where very successful, with PageRank providing better results and the combination beating the best reported results.

When submitting the runs to TREC-2012 our development experiments where not completely finished. Our best run was based on ST expansion alone, and ranked 11th out of the 82 automatic runs. When development finished we were able to improve the PageRank results, and show that both PageRank and ST were improving performance over our baseline system. The best results were those of ST expansion, as submitted to the official task.

In addition, we also report the results when optimizing parameters on the 2012 dataset and evaluating on 2011. The results confirm that both expansion strategies overcome the baseline, with PageRank performing better than ST, and the combination providing the best results. In the future, we plan to perform a thorough analysis of the different queries, in order to learn the reasons for the discrepancy between 2011 and 2012 dataset, and to explore better ways to develop expansion techniques that benefit from the combined expansion approach over medical data.

Acknowledgments

NICTA is funded by the Australian government as represented by Department of Broadband, Communication and Digital Economy, and the Australian Research Council through the ICT Centre of Excellence programme. This research was partially funded by the Ministry of Economy under grant TIN2009-14715-C04-01 (KNOW2 project).

References

[1] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 19–27. Association for Computational Linguistics, 2009.

[2] E. Agirre, M. Cuadros, G. Rigau, and A. Soroa. Exploring knowledge bases for similarity. In N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 2010. European Language Resources Association (ELRA).

[3] E. Agirre, O. L. de Lacalle, and A. Soroa. Knowledge-Based WSD on Specific Domains: Performing better than Generic Supervised WSD. In *Proceedings of IJCAI*, pages 1501–1506, Pasadena, USA, 2009.

[4] E. Agirre and A. Soroa. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 33–41. Association for Computational Linguistics, 2009.

[5] E. Agirre, A. Soroa, and M. Stevenson. Graph-based word sense disambiguation of biomedical documents. *Bioinformatics*, 26(22):2889–2896, Nov. 2010.

[6] T. H. Haveliwala. Topic-sensitive PageRank. In *Proceedings of WWW '02*, pages 517–526, 2002.

[7] Institute of Medicine. 100 initial priority topics for comparative effectiveness research, 2009.

[8] S. Karimi, D. Martinez, S. Ghodke, L. Zhang, H. Suominen, and L. Cavedon. Search for Medical Records: NICTA at TREC 2011 Medical Track. In *Proceedings of the Text Retrieval Conference (TREC)*, 2012.

[9] A. Otegi, X. Arregi, and E. Agirre. Query expansion for ir using knowledge-based relatedness. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 1467–1471, Thailand, 2011.

[10] J. Perez-Iglesias, J. Perez-Aguera, V. Fresno, and Y. Feinstein. Integrating the probabilistic models BM25/BM25F into Lucene. *CoRR*, abs/0911.5046, 2009.

[11] M. Stevenson, E. Agirre, and A. Soroa. Exploiting domain information for word sense disambiguation of medical documents. *JAMIA*, 19(2):235–240, 2012.