Experiments with N-Gram Prefixes on a Multinomial Language Model versus Lucene's off-the-shelf ranking scheme and Rocchio Query Expansion (TEL@CLEF Monolingual Task)

Jorge Machado[1], Bruno Martins[1], José Borbinha[1]
[1] Departmento de Engenharia Informática, Technical University of Lisbon, Portugal.
{jorge.r.machado, bruno.martins, jose.borbinha}@ist.utl.pt

Abstract. We describe our participation in the TEL@CLEF task of the CLEF 2009 ad-hoc track, where we measured the retrieval performance of LGTE, an index engine for Geo-Temporal collection which is mostly based on Lucene, together with extensions for query expansion and multinomial language modelling. We experiment an N-Gram stemming model to improve our last year experiments which consisted in combinations of query expansion, Lucene's off-the-shelf ranking scheme and the ranking scheme based on multinomial language modeling. The N-Gram stemming model was based in a linear combination of N-Gram, with n between 2 and 5, using weight factors obtained by learning from last year topics and assessments. The rochio ranking function was also adapted to implement this N-Gram model. Results show that this stemming technique together with query expansion and multinomial language modeling both result in increased performance.

Keywords: Language Model, Vector Space Model, Lucene, Rocchio QE, Stemming.

1 Introduction

One task of the ad-hoc track at the 2009 edition of the Cross Language Evaluation Forum (CLEF) addresses the problem of searching and retrieving relevant items from collections of bibliographic records from The European Library (TEL@CLEF). Three target collections were provided, each corresponding to a monolingual retrieval task where we participated:

- TEL Catalogue records in English. Copyright British Library (BL)
- TEL Catalogue records in French. Copyright Bibliothèque Nationale de France (BnF)
- TEL Catalogue records in German. Copyright Austrian National Library (ONB)

The evaluation task aimed at investigating the best approaches for retrieval from library catalogues, where the information is frequently very sparse and often stored in unexpected languages.

This paper describes the participation of the Technical University of Lisbon at the TEL@CLEF task. Our experiments aimed at measuring the retrieval performance of the LGTE[1] tool which is implementing the IR service of DIGMAP[2], an EU-funded project which addresses the development of services for virtual digital libraries of materials related to historical cartography [7]. DIGMAP collects bibliographic metadata from European national libraries and other relevant third-party providers (e.g. collections with descriptions available through OAI-PMH), aiming to provide advanced searching and browsing mechanisms that combine thematic, geographic and temporal aspects. In case of success, the ultimate goal of the project is to become fully integrated into The European Library. The LGTE is the DIGMAP text retrieval service which is mostly based on Lucene, together with extensions for using query expansion and multinomial language modeling. A previous version of the system was described in the MSc thesis of Machado [4] and we are now in the process of developing extensions for geo-temporal information retrieval [8].

Like last year in CLEF, we experimented combinations query expansion, Lucene's off-the-shelf ranking scheme and the ranking scheme based on multinomial language modeling, but this year we include an N-Gram model for degraded collections proposed by Parapar in [9]. We adapt this model to our records collections using only N-Gram prefixes instead of the usual sliding window N-Grams. We also perform several experiments on how to use this model in Rochio selection formula for query expansion with encourage results.

2 The experimental environment

The underlying IR system used in our submissions is based on Lucene[3], together with a multinomial language modeling extension developed at the University of Amsterdam and a query expansion extension developed by Neil Rubens. The following subsections detail these components. We adapt our model to use a linear combination of scores using several N-Gram indexes. We also adapt the ranking function defined by Rochio to make use of the N-Gram indexes and the weights assigned to each one of those indexes.

2.1 Lucene's off-the-shelf retrieval model

We started with Lucene's off-the-shelf retrieval model. For a collection D, document d and query q, the ranking score is given by the formula bellow:

$$ranking(q,d) = \sum_{t \in q} \frac{tf_{t,q} \cdot idf_t}{norm_q} \cdot \frac{tf_{t,d} \cdot idf_t}{norm_d} \cdot coord_{q,d} \cdot weight_t \quad (1)$$

[1] http://code.google.com/p/digmap/wiki/LuceneGeoTemporal
[2] http://www.dgmap.eu
[3] http://lucene.apache.org

where:

$$tf_{t,X} = \sqrt{termFrequency(t,X)}, \qquad norm_q = \sqrt{\sum_{t \in q} tf_{t,q} \cdot idf_t^2},$$
$$idf_t = 1 + \log \frac{|D|}{documentFrequency(t,D)}, \qquad norm_d = \sqrt{|d|},$$
$$coord_{q,d} = \frac{|q \cap d|}{|q|} \tag{2}$$

Lucene has been extensively used in previous editions of the CLEF, NTCIR and TREC joint evaluation experiments.

2.2 Lucene extension based on multinomial language modeling

We experimented with a Lucene extension that implements a retrieval scheme based on estimating a language model (LM) for each document, using the formula described by Hiemstra [2]. This extension was developed at the Informatics Institute of the University of Amsterdam[4]. For any given query, it ranks the documents with respect to the likelihood that the document's LM generated the query:

$$ranking(d,q) = P(d \mid q) \propto P(d) \cdot \prod_{t \in q} P(t \mid d) \tag{3}$$

In the formula, d is a document and t is a term in query q. The probabilities are reduced to rank-equivalent logs of probabilities. To account for data sparseness, the likelihood P(t|d) is interpolated using Jelinek-Mercer smoothing:

$$P(d \mid q) = P(d) \cdot \prod_{t \in q} ((1-\lambda) \cdot P(t \mid D) + \lambda \cdot P(t \mid d)) \tag{4}$$

In the formula, D is the collection and λ is a smoothing parameter (in our experiments set to the default value of 0.15). The model needs to estimate three probabilities: the prior probability of the document, *P(d)*; the probability of observing a term in a document, *P(t|d)* and the probability of observing the term in the collection, *P(t|D)*. Assuming the query terms to be independent, and using a linear interpolation of a document model and a collection model to estimate the probability of a query term, the probabilities can be estimated using maximum likelihood estimates:

$$P(t \mid d) = \frac{termFrequency(t,d)}{|d|} \qquad P(d) = \frac{|d|}{\sum_{d' \in D} |d'|}$$
$$P(t \mid D) = \frac{documentFrequency(t,D)}{\sum_{t' \in D} documentFrequency(t',D)} \tag{5}$$

[4] http://ilps.science.uva.nl/Resources/

This language modeling approach has been used in past experiments within the CLEF, NTCIR and TREC joint evaluation campaigns – see for example Ahn et. Al [6].

2.3 N-Gram ranking scheme

The N-Grams stemming technique is very used in corpus resultant from OCR processes because many times the text brings OCR errors. This technique consists in tokenizing the words with a sliding window into tokens of size N, with N assuming several sizes. This process is applied both in documents and queries to increase retrieval performance.

The original N-Grams stemming does not fit very well in our problem because our records were not obtained from OCR processes. On other hand using this technique turns the stemming phase a language independent process, which was our main focus. For that reason, we used a simplistic approach for the N-Grams model which consist in suffixes removal starting in character N+1 and use the prefix for indexing purposes.

Recent experiments related in [9] by Parapar demonstrate that using independent N-Grams indexes, for example from 2 to 5 grams, and combining the individual ranks in a linear combination can improve the results when we find good parameter values to weight each independent index. Our objective was to use this technique. We tokenize our terms in five different ways each of which to create a different inverted file. We create four files with prefixes of N-Grams (2 to 5 grams) and one file with the original terms. The formula to calculate the final score is illustrated by the formula 6 introduced in [9].

$$s(d) = \alpha \times s_{term}(d) + \beta \times s_{5gram}(d) + \gamma \times s_{4gram}(d) + \delta \times s_{3gram}(d) + \epsilon \times s_{2gram}(d) \quad \textbf{(6)}$$

In formula d is the document α, β, γ, δ and ε are the weights assigned to each independent score. To implement this feature in Lucene we re-implement the term scorers of the text models (off-the-shelf and language model) to calculate the score.

The system was trained through experiments with 2008 AdHoc topics and relevance judgments. We found a set of optimal parameter values to weight each inverted file independently. Table 1 shows the optimal values found for each index in each collection. We found that bi-grams worsen the results so we set their weight to zero in the three collections.

Table 1. Descriptions for the eight diferent submited scenarios

Lanuage	*Term*	*5-grams prefix*	*4-grams prefix*	*3-grams prefix*
English	0.45	0,27	0,25	0,03
French	0.53	0,24	0,22	0,01
German	0,55	0,23	0,21	0,01

2.4 Rocchio query expansion

The fact that there are frequently occurring spelling variations and synonyms for any query term degrades the performance of standard techniques for ad-hoc retrieval. To overcome this problem, we experimented with the method for pseudo feedback query expansion proposed by Rocchio [3]. The Lucene extension from the LucQE project5 implements this approach. On test data from the 2004 TREC Robust Retrieval Track, LucQE achieved a MAP score of 0.2433 using Rocchio query expansion.

Assuming that the top D documents returned for an original query qi are relevant, a better query qi+1 can be given by the terms resulting from the formula bellow:

$$q_{i+1} = \alpha \cdot q_i + \frac{\beta}{|D|} \cdot \sum_{d_r \in D} termWeight(d_r) \tag{7}$$

In the formula, α and β are tuning parameters. In our experiments, they were set to the default values of 1.0 and 0.75. The system was trained through experiments with 2008 AdHoc topics and relevance judgments. We found an optimal value of 64 terms for English topics and 40 terms for French and German topics. The terms were extracted from the highest ranked documents (i.e. the |D| parameter) from the original query qi. With the training we obtain optimal values using 7 documents in English and French topics and 8 documents in German topics.

2.5 N-Grams and Rocchio query expansion

In order to deal with N-Gram prefix stemming we need to adapt the Rochio formula. Three techniques were experimented but only the third one improves the results:

- First of all we try to use a list of expansion tokens with a fixed size of tokens of each inverted file (2,3,4,5 Grams and terms), let's say the 15 most relevant tokens of each inverted file. The boosting factors were calculated using the original formula of Rochio to rank terms independently in the different indexes (Inverted File for terms and N-Grams from 2 to 5). To smooth the boosting factor in the expanded query we used the weight of each inverted file (2,3,4,5 Grams and terms) found in training experiments (see Table 1). This doesn't work.
- Second of all we picked up all terms of each top document, we tokenize them to obtain the 2,3,4,5 Grams tokens and we calculate the term relevance using as ranking function of Rochio formula in each inverted file and then we apply the linear combination introduced in section 2.3. This will give the rank of the term. The expanded query was build with the 5 projections of the term, 2-5 Grams tokens and the term, using the ranking calculated with the linear combination as boosting factor. Didn't work at all.
- Third and our best approach which really improves the results was in first place calculate, independently in each inverted file, the score for each

[5] http://lucene-qe.sourceforge.net/

possible N-Grams tokens and terms present in top documents. In second place we order them independently of their source file (2-5Grams or term) and pick the most relevant ones. We calculate the score using Rochio formula for the pairs, source inverted file and token, and we smooth it with the respective weight presented in Table 1 depending on the inverted file. Finally the tokens were ordered by score ignoring their source file and finally the highest scored tokens were used. The score was used as boost factor in final query (e.g.: absolute:information^0.53 index5grams:retrie^0.02, etc).

Our third experiment method turns weak the tokens from less weighted indexes like 2-Grams, and 3-Grams. This fact makes that tokens from weak indexes only were picked if they were very relevant. Expanded queries were mainly composed by tokens of 4-5 Grams and terms. On other hand we presence that all queries had tokens from all indexes. With this technique we deal with all indexes in the same way taking into account that terms from less weighted indexes should be penalized and putted in the same bag.

2.4 Processing the topics and the document collections

Before the actual indexing, the document collections (i.e. the bibliographic records) were passed through the following pre-processing operations:

- **Field Weighting -** The bibliographic records composing the collections from the TEL@CLEF experiment contain structured information in the form of document fields such as *title* or *subject*. We use the scheme proposed by Robertson et. al [5] to weight the different document field according to their importance. Instead of changing the ranking formulas in order to introduce boosting factors, we generate virtual documents in which the content of some specific fields is repeated. The combination used in our experiments is based on repeating the *title* field three times, the *subject* field twice and keeping the other document fields unchanged.
- **Normalization –** The structured documents were converted to unstructured documents for the process of indexing, removing the XML tags and putting the element's contents in separate sentences.

Topic processing was fully automatic and the queries submitted to the IR engine were generated using all parts of the topics (i.e. title, description and narrative). The generation of the actual queries from the query topics was based on the following sequence of processing operations:

- **Parsing and Normalisation -** All characters were reduced to the lowercase unaccented equivalents (i.e. "Ö" reduced to "o" and "É" to "e" etc.) in order to maximise matching.
- **Stop Word Removal -** Stopword lists were used to remove terms that carry little meaning and would otherwise introduce noise. The considered stop words came from the minimized lists distributed with Lucene, containing words such as

articles, pronouns, prepositions, conjunctions or interjections. For English, French and German, these lists contained 120, 155 and 231 terms, respectively.

- **Retrieval –** The resulting queries were submitted to the IR system, which had been used to index the document collections. In some of the submitted runs, variations of the Porter [1] stemming algorithm specific to the language of the collection were used on both the queries and the documents. The stemming algorithms came from the Snowball package[6].

Lucene internally normalizes documents and queries to lower case, also removing stop-words. However, explicitly introducing these operations when processing the topics, has the advantage of facilitating the development of more advanced topic processing (e.g. adding query expansion methods).

3 The experimental story

We submitted 12 official runs to the CLEF evaluation process, a total of 4 runs for each of the languages/collections under consideration in the monolingual task. The runs were selected from those whose obtain best results with the 2008 topics. The conditions under test for each of the submitted runs are as follows:

Table 2. Descriptions for the eight diferent submited scenarios

RUN	*Text Retrieval Model*	*Language*	*Stemmer*	*Query Expansion*
1	LM	EN	Porter (snowball)	Rochio
2	VS	EN	Porter (snowball)	Rochio
3	LM - NGrams	EN	2-5Grams and Term	RochioN-Grams
4	VS - NGrams	EN	2-5Grams and Term	RochioN-Grams
5	LM - NGrams	FR	2-5Grams and Term	No
6	VS - NGrams	FR	2-5Grams and Term	No
7	LM - NGrams	FR	2-5Grams and Term	RochioN-Grams
8	VS - NGrams	FR	2-5Grams and Term	RochioN-Grams
9	LM	DE	Porter (snowball)	Rochio
10	VS	DE	Porter (snowball)	Rochio
11	LM - NGrams	DE	2-5Grams and Term	RochioN-Grams
12	VS - NGrams	DE	2-5Grams and Term	RochioN-Grams

In Table 2 the key LM is the multinomial language model and VS is the Lucene off-the-shelf standard vector space model.

4 Results

Table 3 shows the obtained results for the official runs that make up our TEL@CLEF experiments. The results show that, in terms of the mean average precision (MAP), the weighted N-Grams model outperforms our other submissions. The Rochio query expansion technique together with N-Grams model works fine and

[6] http://snowball.tartarus.org/

improves the results significantly. The weight N-Grams model was better than porter stemming in all situations.

Table 3. Results for the official runs submitted to TEL@CLEF

	English				French				German			
	RUN 1	**RUN 2**	**RUN 3**	**RUN 4**	**RUN 5**	**RUN 6**	**RUN 7**	**RUN 8**	**RUN 9**	**RUN 10**	**RUN 11**	**RUN 12**
num_q	50	50	50	50	50	50	50	50	50	50	50	50
num_ret	50000	50000	50000	50000	50000	50000	50000	50000	50000	50000	50000	50000
num_rel	2527	2527	2527	2527	1853	1853	1853	1853	1559	1559	1559	1559
num_rel_ret	1988	2039	1960	2095	1314	1369	1439	1457	1005	1036	1137	1173
map	0,4143	0,4012	**0,424**	**0,4393**	0,2526	0,2508	**0,2653**	**0,2641**	0,2891	0,281	**0,3049**	**0,3005**
gm_ap	0,254	0,2615	0,2379	0,2401	0	0,1358	0	0,132	0	0	0,1749	0,1648
ndcg	0,6358	0,64	0,6285	0,6432	0,4812	0,5004	0,5073	0,519	0,469	0,4626	0,5242	0,5211
R-prec	0,3953	0,3833	0,4018	0,401	0,2802	0,2635	0,2781	0,2666	0,3092	0,2861	0,3102	0,306
bpref	0,3756	0,3677	0,3897	0,4062	0,2435	0,2301	0,2525	0,2504	0,2689	0,2583	0,2865	0,2943
recip_rank	0,1659	0,198	0,1574	0,1089	0,1243	0,1805	0,1635	0,2157	0,1776	0,154	0,2273	0,1386
P5	0,696	0,664	0,672	0,676	0,496	0,48	0,512	0,476	0,516	0,54	0,524	0,508
P10	0,592	0,556	0,568	0,572	0,408	0,4	0,41	0,388	0,44	0,41	0,416	0,424
P15	0,5307	0,4987	0,496	0,5133	0,3613	0,3427	0,372	0,348	0,3947	0,3747	0,38	0,3773
P20	0,482	0,458	0,462	0,469	0,345	0,319	0,338	0,314	0,359	0,347	0,35	0,337
P30	0,426	0,4033	0,4113	0,4193	0,2987	0,2833	0,2987	0,2773	0,296	0,292	0,2847	0,28
P100	0,2408	0,2326	0,2332	0,2452	0,1624	0,161	0,1652	0,1716	0,1438	0,1406	0,149	0,1506
P200	0,1478	0,1491	0,1453	0,1566	0,0985	0,0996	0,1001	0,109	0,084	0,0811	0,0883	0,0894
P500	0,0728	0,0742	0,073	0,077	0,0473	0,0482	0,0496	0,0517	0,0376	0,0378	0,041	0,0423
P1000	0,0398	0,0408	0,0392	0,0419	0,0263	0,0274	0,0288	0,0291	0,0201	0,0207	0,0227	0,0235

We present now the complete set of experiments using both text models, vector space and language model. We combine all possible situations using rochio query expansion and our different stemming approaches. We demonstrate that these two techniques, stemming and query expansion, improve the results when used alone and even more when combined. We demonstrate that the linear combination of N-Grams is many times better then porter stemming and can be used with rochio query expansion using our term selection method what improves the results even more. Table 4 resumes the obtained results in terms of MAP (Mean Average Precision), P@5 (Precision in first 5 results) and P@10 (Precision in first 10 results) for all possible combinations in the three languages. In French collection the experiment of the rochio query expansion with porter stemming is worst than using just porter stemming, the same is not true with the N-Grams thechnique which inclusively outperfoms all other experiments except language model with porter stemming that is a very strong run, also one of our best runs in 2008 experiments.

Table 4. Results for the official runs submitted to TEL@CLEF

			English			French			German		
Model	Stemm	QE	MAP	P@5	P@10	MAP	P@5	P@10	MAP	P@5	P@10
VS	no	no	0.3403	0.6360	0.5200	0.2030	0.4400	0.3380	0.1357	0.3080	0.2340
LM	no	no	0.3496	0.6480	0.5260	0.2255	0.4680	0.4020	0.1480	0.3160	0.2680
VS	Porter	no	0.3710	0.6320	0.5500	0.2338	0.4360	0.3640	0.2372	0.4920	0.3720
LM	Porter	no	0.3829	0.6800	0.5480	0.2647	0.4760	0.3860	0.2473	0.5040	0.3880
VS	2-5Grams	no	0.3966	0.6760	0.5620	0.2508	0.4800	0.4000	0.2439	0.4800	0.3680
LM	2-5Grams	no	0.3902	0.6800	0.5500	0.2526	0.4960	0.4080	0.2524	0.4880	0.3880
VS	no	Rochio	0.3712	0.6240	0.5400	0.2015	0.4320	0.3420	0.1725	0.3320	0.2740
LM	no	Rochio	0.3778	0.6200	0.5420	0.2213	0.4280	0.3500	0.1921	0.3320	0.3060
VS	Porter	Rochio	0.4012	0.6640	0.5560	0.2186	0.4240	0.3380	0.2810	0.5400	0.4100
LM	Porter	Rochio	0.4143	0.6960	0.5920	0.2391	0.4240	0.3500	0.2891	0.5160	0.4400
VS	2-5Grams	Rochio 2-5Grams	**0.4393**	0.6760	0.5720	**0.2641**	0.4760	0.3880	**0.3005**	0.5080	0.4240
LM	2-5Grams	Rochio 2-5Grams	**0.4240**	0.6720	0.5680	**0.2653**	0.5120	0.4100	**0.3049**	0.5240	0.4160

5 Conclusions

The obtained results support the hypotheses that using Rocchio query expansion together with N-Grams weighted model and a ranking scheme based on language modeling can be beneficial to the CLEF ad-hoc task. The N-Grams prefix stemming linearly combined using tokens of different grams and terms outperform the Porter stemming technique in most scenarios especially when the linguistic stemmers are not appropriate. Using this technique with different text models appear to be independent from those models if the terms score is used independently in the formulas. Unlike last year where our experiments result in poor results both in French and German collections, this year we could obtain very encourage results. Like last year we presence that multinomial language model is almost equal to vector space model in majority of situations. On other hand the multinomial language model has the advantage that we could train it very easily tuning the language model parameters, which was not our objective in this experiment, so we believe that language model has potential to return even better results than vector space model.

References

1. Porter, M. F.: An algorithm for suffix stripping: In: Sparck Jones, K. & Willett, P. (eds.), (1997) Readings in Information Retrieval., pp. 313 - 316. San Francisco: Morgan Kaufmann. (1980)
2. Hiemstra, D.: Using Language Models for Information Retrieval: Ph.D. Thesis, Centre for Telematics and Information Technology, University of Twente. (2001)
3. Rocchio, J. J.: Relevance Feedback in Information Retrieval: In: The SMART Retrieval System. Experiments in Automatic Document Processing: pp 313 - 323. Prentice Hall. (1971)
4. Machado, J.: Mitra: A Metadata Aware Web Search Engine for Digital Libraries: M.Sc. Thesis, Departamento de Engenharia Informática, Technical University of Lisbon. (2008)
5. Robertson, S., Zaragoza, H., and Taylor, M.: Simple BM25 extension to multiple weighted fields: In Proceedings of the Thirteenth ACM international Conference on information and Knowledge Management (Washington, D.C., USA, November 08 - 13, 2004). CIKM '04. ACM, New York, NY, 42-49. (2004)
6. Ahn, D. D., Azzopardi, L., Balog, K., Fissaha, A. S., Jijkoun, V., Kamps, J., Müller, K., de Rijke, M. and Erik Tjong Kim Sang: The University of Amsterdam at TREC 2005: Working Notes for the 2005 Text Retrieval Conference. (2005)
7. Pedrosa, G., Luzio, J., Manguinhas, H., and Martins, B.: DIGMAP: A service for searching and browsing old maps: In Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries (Pittsburgh PA, PA, USA, June 16 - 20, 2008). JCDL '08. ACM, New York, NY, 431-431. (2008)
8. Machado J, Martins B, Borbinha J. (2009), "LGTE: Lucene Extensions for Geo-Temporal Information Retrieval", paper will be presented at the European Conference on Information Retrieval, at Workshop on Geographic Information on Internet, Toulouse, April 2009.
9. Parapar, Javier; Freire, Ana; Barreiro, Álvaro (2009). "Revisiting N-gram Based Models for Retrieval in Degraded Large Collections", European Conference on Information Retrieval, Toulouse, April 2009