Clinical Decision Support with the SPUD Language Model

Ronan Cummins

The Computer Laboratory,
University of Cambridge, UK
ronan.cummins@cl.cam.ac.uk

Abstract. In this paper we present the systems and techniques used by the University of Cambridge for the CDS (Clinical Decision Support) track of the 24^{th} Text Retrieval Conference (TREC). The system was among the best automatic approaches for both CDS tasks and is based on a new language modelling approach implemented using Lucene.[1]

1 Introduction

We outline the main models and techniques used to participate in the CDS track of TREC 2015. The CDS track consisted of retrieving relevant biomedical articles for answering generic clinical questions about medical records. As the documents are full scientific articles they tend to be much longer than the average general Web document. Furthermore, the types of the queries issued for this task are also much longer than those used for typical web search. Therefore, we adopted the use of our new document language model [1] that has been shown to retrieve longer documents more fairly. The recently developed SPUD language model [1] treats document generation using the Pólya process and aims to model word-burstiness directly. It has been shown to incorporate a number of theoretically interesting properties. For example, it models the scope and verbosity hypothesis [3] separately, and reintroduces a measure closely related to inverse document frequency [4]. Therefore, we hypothesised that the SPUD language model would be well-suited to the retrieval of scientific texts. We also studied the performance of a newly developed query modelling technique which re-weights salient terms in longer queries. Furthermore, we investigated query expansion using two different models.

2 Models

We now outline the main models used in our approaches.

[1] https://github.com/ronancummins

2.1 Document Models

We model each document as a mixture of multivariate Pólya distributions. The model captures word burstiness by modelling the dependencies between recurrences of the same word-type. Each document is modelled as follows:

$$\boldsymbol{\alpha}_d = (1-\omega)\cdot\boldsymbol{\alpha}_\tau + \omega\cdot\boldsymbol{\alpha}_c \tag{1}$$

where $\boldsymbol{\alpha}_d$, $\boldsymbol{\alpha}_\tau$, and $\boldsymbol{\alpha}_c$ are the document model, topic model,[2] and background model respectively. The hyper-parameter ω controls the smoothing and is stable at approximately $\omega = 0.8$. Each of these models are multivariate Pólya distributions with parameters estimated as follows:

$$\hat{\boldsymbol{\alpha}}_\tau = \{m_d \cdot \frac{c(t,d)}{|d|} : t \in d\} \qquad\qquad \hat{\boldsymbol{\alpha}}_c = \{m_c \cdot \frac{df_t}{\sum_{t'} df_{t'}} : t \in C\} \tag{2}$$

where m_d is the number of word-types (distinct terms) in d, $c(t,d)$ is the count of term t in document d, $|d|$ is the number of word tokens in d, df_t is the document frequency of term t in the collection C, and m_c is a background mass parameter that can be estimated via numerical methods (see [1] for details). The scale parameters m_d and m_c can be interpreted as beliefs in the parameters $c(t,d)/|d|$ and $df_t/\sum_{t'} df_{t'}$ respectively. For the document collection in the CDS track the parameter m_c is estimated to be 775.

The KL-divergence approach to ranking documents can be used with these document models whereby one takes a point-estimate of the distribution (i.e. $E[\boldsymbol{\alpha}_d]$ is a multinomial) and one can rank documents according to the negative KL-divergence between the query distribution and the expected document multinomial.

2.2 Query Models

When a user formulates a short keyword query (e.g. *hypotension, hypoxia*), it is usually assumed that they have already distilled the topical aspect of the information need. Consequently, one may assume that the probability that a particular query token is topical is 1.0 and this can be normalised accordingly to estimate the maximum likelihood query model (i.e. $\{\frac{1}{2}, \frac{1}{2}\}$). This is the standard method of estimating query models for use with KL-Divergence.

However, when dealing with natural language queries (e.g. *A 44-year-old man with coffee-ground emesis, tachycardia, hypoxia, hypotension and cool, clammy extremities.*) it is likely that some terms are generated by a background query language model. Therefore, we assume that long natural language queries are generated by drawing terms from a query model $\boldsymbol{\alpha}_q$ which consists of both a topical language model $\boldsymbol{\alpha}_{q\tau}$ and a background query language model $\boldsymbol{\alpha}_{qc}$. The topical query model describes the topical information that the user requires,

[2] For the purposes of this paper, we refer to the unsmoothed model as the *topic model* of the document as it explains words not explained by the general background model.

while the background query model describes the syntactic glue of the general query language. Examples of fragments that can be explained by the background query language model are tokens such as *"I, am, looking, for,"*, and *"A, relevant, document, may, include,"* (a stereotypical TREC construct). Therefore, our new query model is defined as follows:

$$\boldsymbol{\alpha}_q = (1 - \lambda_q) \cdot \boldsymbol{\alpha}_{q\tau} + (\lambda_q) \cdot \boldsymbol{\alpha}_{qc} \tag{3}$$

where λ_q is the probability mass of the background query language model. Although the background query language model is likely to contain some structural clues regarding relevance, we simple regard this model as generating noise tokens, and therefore aim to extract the topical part of each query. This can be achieved by determining using Bayes' theorem the probability that a particular query term t was generated by the topical query model as follows:

$$p(\boldsymbol{\alpha}_{q\tau}|t) = \frac{(1 - \lambda_q) \cdot p(t|\boldsymbol{\alpha}_{q\tau})}{(1 - \lambda_q) \cdot p(t|\boldsymbol{\alpha}_{q\tau}) + (\lambda_q) \cdot p(t|\boldsymbol{\alpha}_{qc})} \tag{4}$$

The final step involves determining the distribution of terms in the topical query model $\boldsymbol{\alpha}_{q\tau}$ by normalising over the tokens in q as follows:

$$p(t|\boldsymbol{\alpha}_{q\tau}) = \frac{c(t,q) \cdot p(\boldsymbol{\alpha}_{q\tau}|t)}{\sum_{t' \in q}(c(t',q) \cdot p(\boldsymbol{\alpha}_{q\tau}|t'))} \tag{5}$$

which we call the discriminative query model (DQM). For the specific instantiation of the model using multivariate Pòlya distributions ($\boldsymbol{\alpha}_{q\tau}$), the probability that a particular term t came from the topical part of the query model, when assuming that the background query model is the general collection, is as follows:

$$p(\boldsymbol{\alpha}_{q\tau}|t) = \frac{c(t,q)}{c(t,q) + \frac{(1-\omega_q)}{\omega_q} \frac{df_t}{\sum_{t'} df_{t'}} \frac{m_c \cdot |q|}{m_d}} \tag{6}$$

which can be plugged into Eq. 5 to yield the DQM using the multivariate Pòlya. The one free parameter in this specific query model is ω_q which determines the belief in the background query model. We set this value to be the same as that from the document model such that $\omega = \omega_q$. This model essentially introduces a type of *idf* into longer queries, as common terms occurring in the query will be more likely to come from the background model. As a result, these common terms are down-weighted in the query to varying degrees.

2.3 Psuedo-Relevance Feedback Models

We adopt two types of pseudo-relevance feedback models. Firstly, we use the standard RM3 model [2] with our SPUD framework, where we assume the top $|F|$ documents are relevant. Secondly, we use an approach that extracts the topical terms from the feedback documents in a similar manner to that outlined in the previous section. Given a set of feedback documents F, we then rank terms as follows:

$$p(\boldsymbol{\theta}_Q|t) = \frac{\sum_{d \in F} p(\boldsymbol{\alpha}_\tau|t) \cdot p(q|\boldsymbol{\alpha}_d)}{\sum_{d' \in F} p(q|\boldsymbol{\alpha}_{d'})} \tag{7}$$

where $p(q|\boldsymbol{\alpha}_d)$ is the document score. Given the SPUD language model (Eq. 1) and its parameters estimates (Eq. 2), the probability that the term t was generated from the *topical model* $\boldsymbol{\alpha}_\tau$ of a feedback-document can be calculated via Bayes' theorem as follows:

$$p(\boldsymbol{\alpha}_\tau|t) = \frac{(1-\omega) \cdot \alpha_{\tau_t}}{(1-\omega) \cdot \alpha_{\tau_t} + \omega \cdot \alpha_{c_t}} \tag{8}$$

where α_{τ_t} and α_{c_t} are the parameters of t for the document topic model and background model respectively. A relatively simple intuition for this formula is that topical terms are those that are more likely generated from the topical part of a document than those that are generated by the background model. For all approaches we interpolate 30 feedback terms with the original query using linear-interpolation of 0.5.

3 System and Topics

Our models were all implemented in the Lucene retrieval framework. We stemmed all text using Porter's stemmer and removed a small number of stopwords (the 26 stopwords from the Lucene EnglishAnalyzer).

The topics in the CDS track are of three different types (diagnosis, test, and treatment) depending on what the specific task the clinician is involved in. An example topic is as follows:

```
<topic number="1" type="diagnosis">
<description>A 44 yo male is brought to the emergency room after
multiple bouts of vomiting that has a ‘coffee ground’ appearance.
His heart rate is 135 bpm and blood pressure is 70/40 mmHg.
Physical exam findings include decreased mental status and cool
extremities. He receives a rapid infusion of crystalloid solution
followed by packed red blood cell transfusion and is admitted to
the ICU for further care.
</description>
<summary>A 44-year-old man with coffee-ground emesis,
tachycardia, hypoxia,  hypotension and  cool, clammy extremities.
</summary>
</topic>
```

4 Results

This section presents the results of the six runs submitted to the CDS track (three for task A and three for task B). Table 1 shows details of the six runs submitted.

Table 1. Details of settings used for each run

System	Task	Doc Model	Query Model	Feedback Model	Topic Fields
CAMspud1	A	$\text{SPUD}_{\omega=0.9}$	$\text{DQM}_{\omega=0.9}$	$\text{RM3}_{\lambda=0.5}$, $\lvert F\rvert = 5$	type + summary
CAMspud3	A	$\text{SPUD}_{\omega=0.9}$	$\text{DQM}_{\omega=0.9}$	$\text{QTM}_{\lambda=0.5}$, $\lvert F\rvert = 5$	summary
CAMspud5	A	$\text{SPUD}_{\omega=0.9}$	$\text{DQM}_{\omega=0.9}$	$\text{RM3}_{\lambda=0.5}$, $\lvert F\rvert = 5$	description
CAMspud6	B	$\text{SPUD}_{\omega=0.85}$	$\text{DQM}_{\omega=0.85}$	No Feedback	diagnosis + summary
CAMspud7	B	$\text{SPUD}_{\omega=0.85}$	$\text{DQM}_{\omega=0.85}$	$\text{RM3}_{\lambda=0.5}$, $\lvert F\rvert = 10$	diagnosis + summary
CAMspud8	B	$\text{SPUD}_{\omega=0.85}$	$\text{DQM}_{\omega=0.85}$	$\text{QTM}_{\lambda=0.5}$, $\lvert F\rvert = 10$	diagnosis + summary

4.1 Task A

Table 2 shows the MAP, InfAP, and InfNDCG of the three runs submitted for task A. The rows labelled **median** and **best** show the median and best performance per topic averaged over all 30 topics for all of the runs in the track. It is worth noting that **best** does not represent one system, rather it indicates an upper-bound or oracle approach. All of our approaches performed above the median which is encouraging, with **CAMspud1** being our best run for task A. Our best run is very close in performance to the best single run for task A (labelled **top system**). This approach used the *type* and *summary* fields with RM3 relevance feedback of 30 terms. Overall, our system was the third best for task A (out of 36 automatic systems).

Table 2. MAP, InfAP, and InfNDCG over 30 Topics

System	MAP	InfAP	InfNDCG
CAMspud1	0.1839	0.0758	0.2823
CAMspud3	0.1770	0.0725	0.2791
CAMspud5	0.1678	0.0731	0.2713
top system	-	0.0842	0.2939
median	-	0.0413	0.2038
best	-	0.1258	0.4398

4.2 Task B

Table 3 shows the MAP, InfAP, and InfNDCG of the three runs submitted for task B (where a *diagnosis* field is included in the topics of type treatment and test). For task B we used both the *diagnosis* and *summary* field in all of our runs. Again all of our runs outperform the **median** run. **CAMspud6** does not use pseudo-relevance query expansion and is the worst of the three runs. The only

difference between **CAMspud7** and **CAMspud8** is that the former uses RM3 pseudo-relevance expansion, while the latter uses the new query topic modelling approach. As these results are higher than in task A, it suggests that including the *diagnosis* field is useful. Overall, our system was the fifth best for task B (out of 26 automatic systems).

Table 3. MAP, InfAP, and InfNDCG over 30 Topics

System	MAP	InfAP	InfNDCG
CAMspud6	0.1890	0.0786	0.3059
CAMspud7	0.2232	0.0941	0.3471
CAMspud8	0.2190	0.0912	0.3410
top system	-	0.1140	0.3821
median	-	0.0632	0.2793
best	-	0.1670	0.5348

5 Summary

We experimented with using the new SPUD language modelling approach in the CDS track. In general the approach is quite effective and is among the top five systems on both of the CDS tasks. In summary, we found query expansion to be beneficial, the *summary* field to be more effective than the *description* field, and we found that using the *diagnosis* field (when available) also leads to an improvement in the task.

References

1. Ronan Cummins, Jiaul H. Paik, and Yuanhua Lv. A Pólya urn document language model for improved information retrieval. *ACM Transactions of Informations Systems*, 33(4):21, 2015.
2. Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM.
3. S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc.
4. Karen Spärck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.