Cell Assemblies for Query Expansion in Information Retrieval

Isabel Volpe, Viviane Moreira, and Christian Huyck

***Abstract*— One of the main tasks in Information Retrieval is to match a user query to the documents that are relevant for it. This matching is challenging because in many cases the keywords the user chooses will be different from the words the authors of the relevant documents have used. Throughout the years, many approaches have been proposed to deal with this problem. One of the most popular consists in expanding the query with related terms with the goal of retrieving more relevant documents. In this paper, we propose a new method in which a Cell Assembly model is applied for query expansion. Cell Assemblies are reverberating circuits of neurons that can persist long beyond the initial stimulus has ceased. They learn through Hebbian Learning rules and have been used to simulate the formation and the usage of human concepts. We adapted the Cell Assembly model to learn relationships between the terms in a document collection. These relationships are then used to augment the original queries. Our experiments use standard Information Retrieval test collections and show that some queries significantly improved their results with our technique.**

I. Introduction

Information Retrieval (IR) deals with representation, storage, organization, and access to information items [1]. The IR process starts with a user who has an information need. The user typically translates this need into a query composed of a set of keywords. These keywords are then submitted to an IR System (IRS) (or a search engine) which retrieves the items (textual documents, web pages, images, videos, etc.) that are likely to satisfy the user's information need.

An IR model defines representations for queries and documents and how to match them. A typical IRS pre-processes the text of the documents (tokenization, stopword removal, and stemming) prior to indexing them. This is done offline. When a user query is posed, the IRS applies the same pre-processing to the query text and then proceeds to match the query to the documents. Matching is done by applying a similarity function, which assigns a similarity score to a document d in response to the query q. Such scores are used to generate the ranked list of documents which is returned to the user. This process is depicted in Figure 1.

One of the main challenges in matching a user's query to the relevant documents is that in many cases the keywords chosen by the user as a query are different from the ones chosen by the author writing the document. Two linguistic phenomena contribute to poor results by the IRS: synonymy and polysemy. Synonymy means the same concept can be expressed by different words; e.g. car, automobile and vehicle. Polysemy is the capacity of a word to have multiple meanings; e.g. the word bank may refer to the financial institution, river banks, or memory slots (in a computer). Throughout the years, many approaches have been proposed to solve these problems. Query Expansion (QE) is one of the most widely used methods for solving the synonymy problem. The basic idea is to augment the original query with synonyms and related terms (i.e. more specific or more generic) in order to increase the number of relevant documents retrieved.

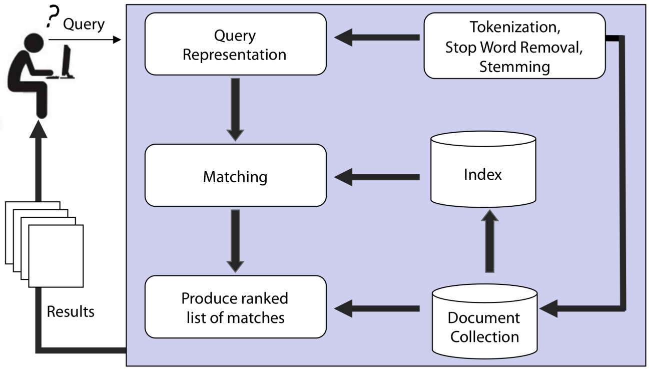
?
Query
Query Representation
Tokenization, Stop Word Removal, Stemming
Matching
Index
Produce ranked list of matches
Document Collection
Results

Fig. 1. Typical architecture of an Information Retrieval System

The main contribution of this paper is to propose a new method for QE based on a Cell Assembly (CA) model. The CA model proposed by Hebb [7] states that groups of active neurons in the brain are responsible for the storage of knowledge in human beings. There is substantial agreement that they form the basis of the human concepts and solve a wide variety of other problems [24]. CAs have been used for several applications [9], [13], [15].

We carried out experiments using a standard IR test collection, the Los Angeles Times (LA Times) collection, which contains 113K news articles from 1994. Our results showed a slight overall improvement. However, for some queries, QE done with our method brought large improvements.

The remainder of this paper is organized as follows: Section II summarizes related work on the topics QE and the use of Neural Networks for IR; Section III introduces the CA model; Section IV presents our proposal of using CAs for QE; in Section V, we report on the experiments and analyze our results; Section VI presents a discussion; finally, Section VII summarizes our work and presents the main conclusions.

II. Related Work

The literature related to our proposed method is divided into two distinct groups: (i) approaches for QE in IR, and (ii) Neural Networks applied to IR. The discussion in this section is focused in these two groups.

Isabel Volpe, Viviane Moreira are with the Institute of Informatics, UFRGS, Brazil,(email: {ivcolpe, viviane}@inf.ufrgs.br). Christian Huyck is at Middlesex University, UK, (email: c.huyck@mdx.ac.uk)

A. Query Expansion

An IRS requires a precise and comprehensive query in order to perform the search and ranking of documents so that only relevant documents are presented to the user. However, the specification of the query is limited by the user's vocabulary and knowledge of the search domain. Query Expansion (QE) aims to retrieve not only documents containing the query terms but also terms that are semantically similar to them.

QE is used to expand the query submitted by the user with terms that have a semantic relationship with those in the original query. The purpose is to enable the retrieval of documents, even if they do not have terms with the same spelling as those that were presented in the original query. What differentiates the types of query expansion is the method by which these additional terms are chosen. The central question is how to generate expanded queries. Two approaches can be taken:

1) *Local methods* use information from the set of documents retrieved by the original query to choose expansion terms. These methods usually require user intervention to mark a few documents they find relevant. This process, known as Relevance Feedback, typically achieves significant retrieval improvements. Experiments by Salton & Buckley [25] on small data collections report improvements from 47% to 160%. However, since Relevance Feedback requires effort from the user, it is not often used in practice [26].
2) *Global methods* expand the query without taking the results retrieved by the original query into consideration. This is usually achieved with the aid of a thesaurus or WordNet [5]. A thesaurus is a controlled dictionary in a given domain of knowledge and is used for identifying synonymous expressions and linguistic entities that are semantically similar. The basic procedure is: for each term in a query, automatically expand with synonyms and related terms from the thesaurus. The main advantage of this method is not requiring user intervention. The main limitation is that building a thesaurus manually is very costly [18], [28]. WordNet [5] is organized in sets of synonyms with the terms of same meaning allowing searches for semantically related nodes. Grootjen et al. [6] and Parapar et al. [20] used WordNet as a source of additional terms to complement the user's query. Both studies report that no significant improvement was obtained with the expansion.

B. Neural Network Models for IR

In this section we present some studies that use Artificial Neural Networks (ANNs) as an alternative IR model. The main motivation for the use of ANNs in IR is that they perform well at pattern matching tasks, and this ability can potentially aid document retrieval.

One of the first attempts to use an ANN for IR was done by Belew [2] who proposed a three-layer network with authors, index terms and documents. His system employed relevance feedback from the users to generate a consensual representation of the meaning of keywords and documents shared by the group of users. Kwok [16] also used a three-layer network. The idea was to reformulate the probabilistic model for IR. Queries, index terms and documents were represented by neurons in different layers. Query processing was done by externally activating the neurons that were present in the query (first layer). These neurons would spread activation to the index terms in the middle layer, which in turn would spread their activation to the documents in the third layer. The level of activation of each document neuron was used to rank the documents in relation to the query. No experiments were reported in the work. Similarly, Wilkinson & Hingston [27] used a three-layer ANN representing queries, index terms, and documents. This study performed standard IR experiments on small text collections. Ranking using the proposed ANN achieved an improvement of 14% in relation to the cosine measurement.

Boughanem et al. [3] applied a two-layer ANN for performing pseudo-relevance feedback during Trec-7. This study reports a loss of 5% in average precision compared to the baseline run which did not include query reformulation using an ANN.

In preliminary studies, Huyck & Orengo [11] have shown that CAs can be used to perform categorization and IR, indicating its potential for improving performance when used in an IRS. However, this study was exploratory and a series of questions about the use of CAs in IR remain unanswered.

More recently, Roberson & Dankel [23] used a Morphological NN (MNN), which differs from other neural networks by the way processing occurs in the nodes. Multiplication and addition are replaced by addition and maximum (or minimum), respectively. As a result MNN computation is nonlinear before thresholding. Retrieval experiments on the TIME test collection reported results which were significantly worse than the vector space model.

Desjardins et al. [4] proposed an auto-associative NN to perform matching between queries and documents. In this type of network, all nodes are inter-connected and the synaptic weights are adjusted by a Hebbian rule. The authors report on experiments in which 2,000 documents and seven queries were selected from the TREC FT943 collection. The results showed that the auto-associative NN outperformed the vector space model at low recall levels.

Existing work on the use of ANNs for IR have either not reported on experimental results [2], [16], or have used very small test collections [4], [11], [23], [27], or failed to produce improvements [3]. Thus, over twenty years after the first studies were published, the applicability of ANNs for IR is still an open question. In this paper, we aim to move a step towards finding an answer by applying a different type of ANN to IR and by carrying out experiments on a test collection used in standard IR evaluations.

III. Cell Assembly Model

The model of CAs proposed by Hebb [7] suggests that groups of neurons in the brain are responsible for storing knowledge in human beings. CAs are groups of neurons that have substantial internal synaptic strength. Through network reverberation, the CA can remain active even after the external stimulus has ceased [8]. By specifying the network topology using a series of texts, we can learn relationships between terms.

The CA Network model [8] is a neural model, where CAs emerge via learning. The model has been used for IR [11], learning hierarchical categories [12], and as a video game agent [10], among other applications. Unlike most other ANNs, this is a model of mammalian neural processing, albeit a relatively coarse neural model.

The processing is broken into discrete time steps. On each time step, any neuron that has an activation level greater than a given threshold will fire, and the activation levels for all of its post-synaptic neurons are updated. Neurons are connected by a small number of synapses. In the simulations below, neurons in the network are connected to 40 other neurons [11]. Figure 2 shows the network as a rectangular matrix. The neurons that are firing are shown in blue.

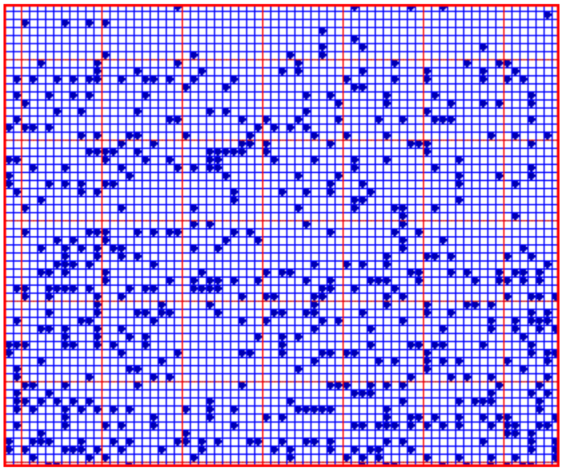

Fig. 2. Snapshot of the network's activation. The cells in blue represent a neuron firing.

The model was designed to have an operation similar to that of natural neural systems. At any given run of the network, this should lead to activation of the appropriate CA.

The CA Network Model uses fatiguing Leaky Integrate and Fire (fLIF) neurons described briefly in section III-A (and more completely in [12]). The neurons are connected by synapses that learn according to Hebbian rules described briefly in section III-B (and more completely in [12]). Neurons are loosely connected so that they do not have connections to most other neurons.

A. Neurons

The CA Network simulator is based on fLIF neurons. Neurons receive activation from other neurons via synaptic connections. A neuron fires only if it has sufficient activation to exceed the activation threshold. When a neuron fires, it sends activation proportional to the synaptic weight through its synapses to other neurons. Thus, they have five key neural properties, the first three are fairly common in ANN models, but the latter two are less common [8]:

1) *Connection Strength*: neurons have connections to other neurons, connections may have positive or negative strength and the continuous activation is simulated by time steps.
2) *Activation*: a neuron has an activation level that is largely based on the firing behavior of neurons that connect to it.
3) *Activation Threshold*: a neuron fires if it has enough activation to surpass the activation threshold.
4) *Decay*: if a neuron does not fire, it retains its activity, but that activity decays. New input can lead to a net gain of activation. Decay is a constant that applies to active and inactive nodes.
5) *Fatigue*: the more time that the neuron is active the higher the threshold becomes. This neuron fatigues and this will make it less likely to fire.

The activation (property 2) that neuron has at a time step uses the weight of the synapses (property 1), and is described by Equation 1 [21]. This uses decay (property 4) directly. If it has enough energy, a neuron will fire and spread its activity (property 3).

$$h_i^t = \frac{h_i^{t-1}}{d} + \sum_{j \in V} w_{ji}, 1 > d \tag{1}$$

The current activation h for a neuron i at a time t is the activation from the last time step divided by a decay factor d plus the new activation coming in. This new activation is the sum of the inputs of all neurons $j \in V$, weighted by the value of the synapse from neuron j to neuron i; V being the set of all neurons that are connected to i that fired in time step $t-1$. However, if the neuron fired in the prior time step, no activation carries over from $t-1$.

A neuron should not be able to continually fire repeatedly. Thus the model also uses neural fatigue (property 5). This is modeled by raising the activation threshold each time a neuron fires. When a neuron does not fire, its activation threshold decreases at each time step until it reaches a given base level [8].

B. Hebbian Learning

The CA model uses a Hebbian Learning rule. The general Hebbian learning rule states that if two neurons fire at the same time step, the strength of their synapses should be increased. The two neurons are called pre-synaptic and post-synaptic, and energy flows from the pre-synaptic neuron to the post-synaptic neuron when the pre-synaptic neuron fires.

In order to prevent saturation of the synapse, the model also uses an anti-Hebbian rule that decreases the weight of the synapse each time the pre-synaptic neuron fires and the post-synaptic neuron does not fire.

For this paper, a compensatory learning rule is used. This rule limits the total synaptic strength of a neuron. As a result, neurons with low correlations will have their influence increased and neurons with high correlations will have their influence reduced. This rule is similar to IDF (inverse document frequency) weighting widely used in IR (see [12] for more detail).

IV. Cell Assemblies for Query Expansion

In order to adapt the CA model to perform QE, it was necessary to disregard some of the constraints that are important when modeling the mammalian brain. For example, neurons are likely to be connected to nearby neurons in the brain and not to distant ones; this constraint has been ignored in these simulations. The most important difference is that each term is represented by a single neuron which is not biologically plausible since terms are represented by many neurons across many brain areas.

In the CA Network model, each term is represented by a neuron and, the documents are represented by the set of neurons aswsociated with the words contained in the document. By representing a term using one neuron, the size of the network is markedly reduced enabling the encoding of several thousand terms with a network of several thousand neurons. The documents are indexed and faced with the task of comparing the query with these sets of terms. Each term that occurs in more than one document was assigned a neuron. Each neuron is connected to other neurons representing the terms that it co-occurs with at least once across the document collection. Between these possible connections, the connections are selected randomly.

A. Training Phase

This is the first step of the process, in which documents from the collection are presented to the CA network. The neurons corresponding to terms that are present in the documents receive external activation.

As the documents are presented to the system, the weights of the synapses between pre- and post-synaptic neurons are adjusted. The result of this adjustment leads to weights that should reflect the co-occurrence relationship of the terms in the documents. The learning process occurs by synaptic modification, using the compensatory learning rule [11].

At the end of the learning process, the semantic relationships among terms are modeled. These semantic relations are derived from characteristics of the distribution of terms in the document collection. This method is based on the hypothesis that term co-occurrence statistics provide useful information about the semantic relationships between terms.

B. Querying Phase

The queries consist of a set of keywords that describe the information needs of the user. During this phase, the queries are presented to the CA network. The neurons that represent the terms in the query receive external stimulation. As a result, they fire and send activation to other neurons through synaptic connections. External stimulation and activation propagation continues for a few cycles and the state of the network is saved. This process will have the effect of expanding the original query with correlated terms. The rationale is that by adding correlated terms, the IR system will retrieve more relevant documents.

V. Experiments

The aim of the experiments was to assess the feasibility of employing CAs for QE. This section details the experimental setup and the results obtained.

A. Experimental Setup

The test collection used was the Los Angeles Times, which is composed of news articles published in 1994 and is part of the CLEF test suite[1]. Table I shows details of the document collection.

TABLE I
Characteristics of the Test Collection

Number of documents	113,005
Number of queries	50
Number of unique stems	94,027
Number of relevant docs	821

The 50 query topics are those used in the CLEF IR evaluation campaign. Topics 91 to 140 were used. Table II shows an example of a query topic. Each topic has an id, a title, a description, and a narrative. As it is normally done in IR, our queries were composed of terms from the title and the description.

TABLE II
Example of a query topic.

```
<top>
<num> C140 < /num>
<EN-title> Mobile phones < /EN-title>
<EN-desc> Prospects for the use of cellular phones. < /EN-desc>
<EN-narr> Relevant documents report on the prospects for the
use of cellular phones and the development of the mobile phone
industry. < /EN-narr>
< /top>
```

We used the Porter stemmer [22] to remove word suffixes, and we also removed stopwords according to the list provided by SMART[2]. The IR system we used was Zettair[3], which is a compact and fast search engine developed by RMIT University (Australia). It performs a series of IR tasks such as indexing and matching. Zettair implements several methods for ranking documents in response to queries. We used Okapi BM25 [14] as it showed the best results in preliminary experiments. Given a query Q containing a set of keywords $q1, ...q_n$, the BM25 score of a document d is given by Equation 2.

[1] http://www.clef-campaign.org/

[2] ftp://ftp.cs.cornell.edu/pub/smart/english.stop

[3] http://www.seg.rmit.edu.au/zettair/

$$BM25(d,Q) = \sum_{i=1}^{n} IDF(q_i) \frac{tf(q_i,d)\cdot(k_1+1)}{tf(q_i,d)+k_1(1-b+b\frac{|D|}{avg})} \quad (2)$$

where $tf(q_i, d)$ is the frequency of term q_i in document d, $|d|$ is the length (in words) of document d, $avgdl$ is the average document length in the collection, k_1 and b are parameters that tune the importance of the presence of each term in the query and the length of the documents, and $IDF(q_i) = \log \frac{N}{n(q_i)}$ where N is the total number of documents in the collection, and $n(q_i)$ is the number of documents containing q_i. In our experiments, we used $k_1 = 1.2$ and $b = 0.75$.

Trec_Eval[4] was used to evaluate the results. It is the standard tool used by the TREC campaign for evaluating an ad hoc retrieval run, given the results file and a standard set of relevance judgments. For each query, the top 1,000 documents were retrieved.

B. Quality Measures

IR evaluation is based on Precision and Recall. Precision (Eq. 3) is the portion of the retrieved documents which is actually relevant to the query. Recall (Eq. 4) is the fraction of the relevant documents which is retrieved by the IRS.

$$Precision(P) = \frac{\#Relevant \bigcap \#Retrieved}{\#Retrieved} \quad (3)$$

$$Recall(R) = \frac{\#Relevant \bigcap \#Retrieved}{\#Relevant} \quad (4)$$

However, Precision and Recall are set based measures, therefore, they do not take into consideration the ordering in which the relevant items were retrieved. For this reason, the most widely used measure to assess the quality of ranked retrieval results for a given query is the *average precision* (AvP). AvP emphasizes returning more relevant documents earlier in the ranking. For a set of queries, we calculate its *Mean Average Precision* (MAP) according to Equation 5 [18].

$$MAP(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j} P(R_{jk}) \quad (5)$$

where $|Q|$ is the number of queries, R_{jk} is the set of ranked retrieval results from the top result until document d_k, and m_j is the number of relevant documents for query j.

A graphical way to compare the results by different experimental runs is the Precision-Recall curve. To build these curves, we compute the interpolated precision at 11 fixed recall points $0, 0.1, ..., 1$.

C. Experimental Runs

For the baseline run, we simply submitted the original text from the topics as queries. For the QE run, the following procedure was taken:

1) *Building the CA network:* We built a network by connecting each neuron to 40 other randomly selected neurons which it co-occurs with in at least one document. All initial weights were set to 0.1.
2) *Training the CA network:* Among the 821 relevant documents, 398 were randomly selected to be used in training. In order to have a fair evaluation of the benefit brought by QE, these documents were removed from the evaluation and were not indexed by Zettair. During the training phase, each document was presented to the CA network for 1 cycle. This was repeated 20 times. At the end of this process the synaptic weights have been set. The parameters used in the training were as follows:
   - Activation threshold 0.80
   - Connectivity 0.90
   - Decay 2.0
   - Fatigue 0.2
3) *Presenting the queries:* The neurons corresponding to the original query terms were externally stimulated. We let the activation spread for five cycles and then saved the status of the network. The terms corresponding to the neurons that were active after five cycles of spreading activation were added to the original query.

D. Results

To illustrate an example of expansion, let us take query topic 140 (shown in full at Table II). For this query, the baseline run was composed by the following nine stems: *cellular, develop, docum, industry, mobil, phone, prospect, relev,* and *report*. The QE run for this query had 119 stems in total including many related terms such as: *bellsouth, motorola, technolog, telecommun,* and *telephon*.

To evaluate the QE run in comparison to the baseline, we used the quality measures described in Section V-B. Table III shows a summary of our results in terms of MAP. The Recall-Precision curves in Figure 3 show that the experimental run in which the CAs were used to expand the queries was superior to the baseline at low recall levels ($\leq$ 0.3). This means that the expansion enabled retrieving relevant documents earlier in the ranking.

TABLE III
SUMMARY OF RESULTS

	Baseline	**QE via CAs**
Mean Average Precision	27.80%	28.46%
Relevant Retrieved	372	360
Avg. query terms	18.84	29.52

In terms of MAP, the improvements brought by the expansion are slight (2.36%). Thus, a topic-by-topic analysis was performed. This analysis consisted in assessing how

[4] http://trec.nist.gov/trec_eval/

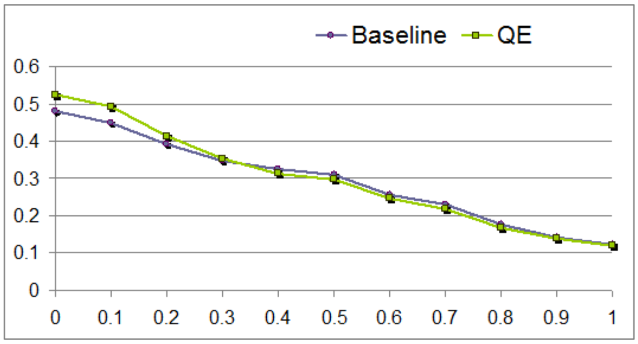

Fig. 3. Recall-Precision Curves for LA Times

many topics improved or worsened with QE via CAs. The results of the analysis show that 20 topics have improved, 17 topics worsened and 13 had no difference. As a general tendency, the queries that had the largest improvements were the ones with few relevant documents. Table IV shows the top ten topics which were helped by use of CAs and the ten topics that were most harmed. The last column shows the proportional gain or loss from the QE run in relation to the baseline. We observed that the gains outweighs the losses.

The topic with the highest gain was 129. For this topic, the relevant documents in the Baseline run were at ranks 3, 22, 103, 109, and 121. In the QE run, these relevant documents were at ranks 1, 20, 71, 87, and 116. This means that query representation improved and thus the relevant documents have been retrieved earlier. In all queries in which the baseline had a perfect retrieval score (i.e., MAP = 1), QE also obtained the same result.

The topic with the highest loss was 116. The QE run failed to retrieve one of the relevant documents. Also, the ones that were retrieved were at lower ranks in comparison to the Baseline run.

E. Training Individual Topics

One hypothesis for QE improving some queries while harming others is that we were trying to teach too many different concepts at once to the CA network. Bear in mind that our document collection has one year of news articles dealing with a very broad range of subjects including politics, sports, culture, science, and entertainment. If that were true, teaching fewer concepts at a time would yield more improvement. To test this hypothesis, we picked ten queries (five among the ones with the biggest improvements and five among the ones with the biggest losses) and applied the process described in Section V-C (i.e., building the network, training, and querying) to generate the QE run.

The results are shown in Figure 4 and Table V. The Precision-Recall curves demonstrate that nearly all topics which were improved in the group training had an even larger improvement with the individual training. Topic 129 had a remarkable improvement, going from 10% on the Baseline to 66% with individual training. Individual training also helped topics 114 and 116 which had losses in the group training. These improvements support our hypothesis that teaching fewer concepts brings more improvements. However, queries 95, 103, and 119 even with individual training score less than the baseline. Therefore, we conclude that topics will respond differently to QE methods. A deeper analysis of those topics has shown that they contain terms that are very frequent in the news collection. For example, in query 103, the rarest term appeared in 448 documents. The other two topics follow the same tendency. In these cases, the expansion will not be helpful since the added terms will end up retrieving even more documents which can reduce the rank of the truly relevant documents. In contrast, queries with infrequent words have the largest improvements. For example, topic 94 which more than doubled its MAP with QE, had terms appearing in only 31 documents across the entire collection.

TABLE IV

TEN TOPICS WITH THE BIGGEST INCREASE IN MAP

Topics with Largest Gains				
Topic	**Baseline**	**QE**	**Change**	**% Change**
129	0.1063	0.2463	+0.14	+132%
126	0.0593	0.1272	+0.07	+115%
133	0.1563	0.2837	+0.13	+82%
94	0.2829	0.4554	+0.17	+61%
91	0.2418	0.3038	+0.06	+26%
121	0.4809	0.5950	+0.11	+24%
140	0.1874	0.2262	+0.04	+21%
99	0.0663	0.0799	+0.01	+21%
135	0.0970	0.1100	+0.01	+13%
131	0.2906	0.3215	+0.03	+11%

Topics with Largest Losses				
Topic	**Baseline**	**QE**	**Change**	**% Change**
116	0.3097	0.1615	-0.15	-92%
122	0.1224	0.0770	-0.05	-59%
124	0.2477	0.1691	-0.08	-46%
95	0.3317	0.2775	-0.05	-20%
103	0.4403	0.3855	-0.05	-14%
114	0.4072	0.3784	-0.03	-8%
123	0.3459	0.3243	-0.02	-7%
119	0.7712	0.7374	-0.03	-5%
120	0.3916	0.3771	-0.01	-4%
92	0.7013	0.6876	-0.01	-2%

TABLE V

AVERAGE PRECISION FOR QUERIES TRAINED INDIVIDUALLY

Topics with Gains					
	Baseline	**QE**	**QE-Ind.**	**Change**	**% Change**
91	0.2418	**0.3038**	0.2963	0.05	23%
94	0.2829	0.4554	**0.6002**	0.32	112%
129	0.1063	0.2463	**0.6624**	0.56	523%
133	0.1563	0.2837	**0.4299**	0.27	175%
140	0.1874	0.2262	**0.2266**	0.04	21%
MAP	**0.1949**	**0.3032**	**0.4431**	**0.25**	**127%**

Topics with Losses					
	Baseline	**QE**	**QE-Ind.**	**Change**	**% Change**
95	**0.3317**	0.2775	0.3184	-0.01	-4%
103	**0.4403**	0.3855	0.3020	-0.14	-31%
114	0.4072	0.3784	**0.4435**	0.04	9%
116	0.3097	0.1615	**0.3294**	0.02	6%
119	**0.7712**	0.7374	0.6006	-0.17	-22%
MAP	**0.4520**	**0.3881**	**0.3988**	**-0.05**	**-12%**

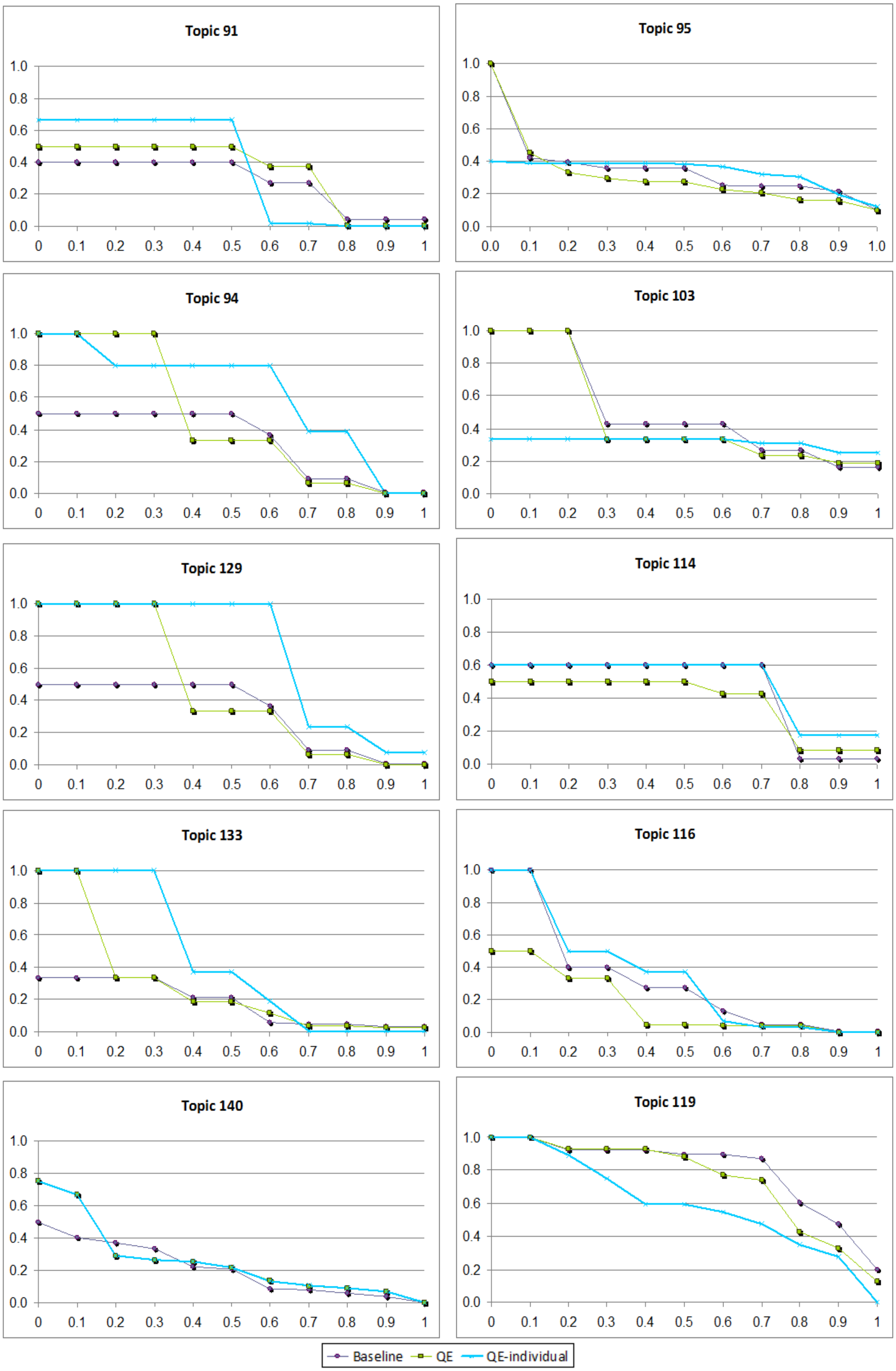

Fig. 4. Precision-Recall curves for the individually trained topics. The first column contains topics which were improved by QE while the second column contains the topics that were harmed by QE.

VI. Discussion

The conclusions drawn from our experiments corroborate findings from other studies. In IR, different queries are better solved by different approaches. Mandl & Womser-Hacker [17] demonstrated this fact when evaluating several CLEF runs. They observed a high standard deviation for the performance of the topics and a high standard deviation for the performance of each run. Their conclusion is that no run performed well in all topics. In the same line, Orengo & Huyck [19] have shown that the main source of impact on the change in performance produced by Relevance Feedback is the topics. However, the authors did not report the characteristics of the topics which made them respond differently. In this paper we identified that the frequency of the query terms in the documents is one such characteristic.

We cannot do a direct comparison between our results and results from other neural models applied to IR because they all use different test collections. Still it is worth reporting their findings. Boughanem et al. [3] mention a loss in MAP on the TREC collections. Their baseline had 27% while their approach scored 22.78%. Roberson & Dankel [23] also reported losses in comparison to the vector-space model in the TIME collection. Desjardins et al. [4] achieved higher precision at low recall levels (<0.3). However, there were only seven queries in their study and the collection contained just 2K documents. Similarly, Huyck & Orengo [11] report on improvements on the Cranfield test collection, which is also very small. Considering results from these other studies, our method performed well. The overall improvement was modest, but some queries more than doubled their scores.

VII. Conclusions

This paper proposed a query expansion method based on the Cell Assembly model. The rationale was to model the relationships between terms using Hebbian learning and then use these learnt relationships to augment original queries with related terms.

Our method was evaluated using the standard information retrieval methodology, using a collection with 113K documents and 50 query topics. The results show that the overall improvement was small (2.36%). However, some queries had significant enhancements more than doubling their performances. An in depth analysis of the results concluded that characteristics of the query topics affect how they respond to the expansion. Topics with very frequent terms in the document collection tend to be harmed by the expansion, while topics with rare terms tend to improve.

VIII. Acknowledgments

This work was partially supported by CNPq (Brazil) processes 550763/2009-0 and 305022/2008-3.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, —1999—.

[2] R. K. Belew. Adaptive information retrieval: using a connectionist representation to retrieve and learn about documents. *SIGIR Forum*, 23:11–20, —1989—.

[3] M. Boughanem, C. Julien, J. Mothe, and C. Soule-Dupuy. Mercure at trec7. In *TREC-7*, —1998—.

[4] G. Desjardins, R. Proulx, and R. Godin. An auto-associative neural network for information retrieval. In *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, pages 3492 –3498, 2006.

[5] C. Fellbaum. Wordnet - an eletronic lexical database. mit press, —1998—.

[6] F. A. Grootjen and T. P. V. d. Weide. Conceptual query expansion. *Data Knowl. Eng.*, 56(2):174–193, —2006—.

[7] D. Hebb. *The Organization of Behaviour: A neuropsycological theory*. Ed. Wiley, New York, —1949—.

[8] C. Huyck. Modelling cell assemblies, —1999—.

[9] C. Huyck. Overlapping cell assemblies from correlators. *Neurocomputing*, Volume 56:435–439, —2004—.

[10] C. Huyck and E. Byrne. CABot1: Technical report, —2009—.

[11] C. Huyck and V. Orengo. Information retrieval and categorisation using a cell assembly network. *Neural computing & applications*, 14(4):282–289, —2005—.

[12] C. R. Huyck. Creating hierarchical categories using cell assemblies. *Connect. Sci*, 19(1):1–24, —2007—. 1392501.

[13] J. E. Ivancich, C. R. Huyck, and S. Kaplan. Cell assemblies as building blocks of larger cognitive structures. *Behavioral and Brain Sciences*, pages pp. 292–293, —1999—. 10.1017/S0140525X99331824.

[14] K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Inf. Process. Manage.*, 36:779–808, November 2000.

[15] A. Knoblauch, R. Kupper, M. Gewaltig, U. Korner, and E. Korner. A cell assembly based model for the cortical microcircuitry. *Neurocomputing*, 70:1838–1842, 2007.

[16] K. L. Kwok. A neural network for probabilistic information retrieval. *SIGIR Forum*, 23(SI):21–30, —1989—. 75338.

[17] T. Mandl and C. Womser-Hacker. Linguistic and statistical analysis of the clef topics. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *CLEF*, volume 2785 of *Lecture Notes in Computer Science*, pages 505–511. Springer, 2002.

[18] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, —2008—. 1394399.

[19] V. M. Orengo and C. R. Huyck. Relevance feedback and cross-language information retrieval. *Information Processing & Management*, 42(5):1203–1217, 2006.

[20] D. Parapar, A. Barreiro, and D. E. Losada. Query expansion using wordnet with a logical model of information retrieval. In *Proceedings of IADIS*, pages 487–494, —2005—.

[21] P. Passmore and C. Huyck. Models of cell assembly decay. In *Cybernetic Intelligent Systems, 2008. CIS 2008. 7th IEEE International Conference on*, pages 1 –6, sept. 2008.

[22] M. Porter. An algorithm for suffix stripping. *Program*, —1980—.

[23] C. Roberson and D. D. D. II. A morphological neural network approach to information retrieval. In *FLAIRS Conference*, pages 184–185, 2007.

[24] Y. Sakurai. The search for cell assemblies in the working brain. *Behavioural Brain Research*, 91(1-2):1–13, —1998—.

[25] G. Salton and C. Buckley. *Improving retrieval performance by relevance feedback*. Morgan Kaufmann Publishers Inc., —1997—.

[26] A. Spink. Term relevance feedback and query expansion: Relation to design. In *SIGIR*, pages 81–90, 1994.

[27] R. Wilkinson and P. Hingston. Using the cosine measure in a neural network for document retrieval. In *SIGIR*. ACM, —1991—.

[28] J. Zhang, B. Deng, and X. Li. Concept based query expansion using wordnet. In *Proceedings of the 2009 International e-Conference on Advanced Science and Technology*, AST '09, pages 52–55, Washington, DC, USA, 2009. IEEE Computer Society.