Passage Scoring for Question answering via Bayesian inference on lexical relations

Deepa Paranjpe, Ganesh Ramakrishnan, Sumana Srinivasan
{adeepa,hare}@cse.iitb.ac.in, sumana@it.iitb.ac.in
Dept. of Computer Science and Engg.,
Indian Institute of Technology, Mumbai, India

Abstract

Many researchers have used lexical networks and ontologies to mitigate synonymy and polysemy problems in Question Answering (QA), systems coupled with taggers, query classifiers, and answer extractors in complex and ad-hoc ways. We seek to make QA systems reproducible with shared and modest human effort, carefully separating knowledge from algorithms. To this end, we propose an aesthetically "clean" Bayesian inference scheme for exploiting lexical relations for passage-scoring for QA . The factors which contribute to the efficacy of Bayesian Inferencing on lexical relations are *soft word sense disambiguation*, *parameter smoothing* which ameliorates the data sparsity problem and *estimation of joint probability over words* which overcomes the deficiency of naive-bayes-like approaches.

1 Introduction

This paper describes an approach to probabilistic inference using lexical relations, such as expressed by a WordNet, an ontology, or a combination, with applications to passage-scoring for open-domain question answering (QA).

The use of lexical resources in Information Retrieval (IR) is not new; for almost a decade, the IR community has considered the use of natural language processing techniques (Lewis and Jones, 1996) to circumvent synonymy, polysemy, and other barriers to purely string-matching search engines. In particular, a number of researchers have attempted to use the English WordNet to "bridge the gap" between query and response. Interestingly, the results have mostly been inconclusive or negative (Fellbaum, 1998a). A number of explanations have been offered for this lack of success, some of which are

- presence of unnecessary links and absence of necessary links in the WordNet (Fellbaum, 1998b),
- hurdle of Word Sense Disambiguation (WSD) (Sanderson, 1994)
- ad-hocness in the distance and scoring functions (Abe et al., 1996).

2 Proposed approach

2.1 An inferencing approach to QA

Given a question and a passage that contains the answer, how do we correlate the two ? Take for example, the following question

> What type of animal is Winnie the Pooh?

and the answer passage is

> A Canadian town that claims to be the birthplace of Winnie the Pooh wants to erect a giant statue of the famous bear; but Walt Disney Studios will not permit it.

It is clear that there is a linkage between the question word *animal* and the answer word *bear*. That the word *bear* occurred in the answer, in the context of Winnie, means that there was a hidden "cause" for the occurrence of *bear*, and that was the concept of { animal}.

In general, there could be multiple words in the question and answer that are connected by many hidden causes. The causes themselves may have hidden causes associated with them. These causal relationships are represented in ontologies and WordNets. The familiar English WordNet, in particular, encodes relations between words and concepts. For instance WordNet gives the *hypernymy* relation between the concepts { animal} and { bear}.

2.2 WordNet

WordNet (Fellbaum, 1998b) is an online lexical reference system in which English nouns, verbs, adjectives and adverbs are organized into synonym sets or *synsets*, each representing one underlying lexical concept. Noun synsets are related to each other through *hypernymy* (generalization), *hyponymy* (specialization), *holonymy* (whole of) and *meronymy* (part of) relations. Of these, (*hypernymy*, *hyponymy*) and (*meronymy*,*holonymy*) are complementary pairs.

The verb and adjective synsets are very sparsely connected with each other. No relation is available between noun and verb synsets. However, 4500 adjective synsets are related to noun synsets with *pertainyms* (pertaining to) and *attra* (attributed with) relations.

For example, the synset { dog, domestic_dog, canis_familiaris} has a hyponymy link to { corgi, welshcorgi} and meronymy link to { flag} ("a conspicuously marked or shaped tail"). While the hyponymy link helps us answer the question (TREC#371) "A corgi is a kind of what?", the meronymy connection here is perhaps more confusing than useful: this sense of *flag* is rare.

2.3 Inferencing on lexical relations

It is surprisingly difficult to make the simple idea of bridging passage to query through lexical networks perform well in practice. Continuing the example of Winnie the bear (section §2.1), the English WordNet has five synsets on the path from *bear* to *animal*: {carnivore...}, {placental_mammal...}, {mammal...}, {vertebrate..}, {chordate...}.

Some of these intervening synsets would be extremely unlikely to be associated with a corpus that is not about zoology; a common person would more naturally think of a bear as a kind of animal, skipping through the intervening nodes.

It is, however, dangerous to design an algorithm which is generally eager to skip across links in a lexical network. E.g., few QA applications are expected to need an expansion of "bottle" beyond "vessel" and "container" to "instrumentality" and beyond. Another example would be the shallow verb hierarchy in the English WordNet, with completely dissimilar verbs within very few links of each other. There is also the problem of missing links.

Another important issue is *which 'hidden causes'* (synsets) should be inferred to have caused words in the text. This is a classical problem called word sense disambiguation (WSD). For instance, the word *dog* belongs to 6 noun synsets in WordNet. Which of the 6 synsets should be treated as the 'hidden cause' that generated the word *dog* in the passage could be inferred from the fact that *collie* is related to *dog* only through one of the latter's senses - it's sense as {dog, domestic dog, Canis_familiaris}. But this problem of finding the 'appropriate' hidden causes, in general, in non-trivial. Given that state-of-the-art WSD systems perform not better than 74% (Sanderson, 1994) (Lewis and Jones, 1996) (Fellbaum, 1998b), in this paper, we use a probabilistic approach to WSD - called 'soft WSD' (Pushpak,) ; hidden nodes are considered to have probabilistically 'caused' words in the question and answer or in other words, causes are probabilistically 'switched on'.

Clearly, any scoring algorithm that seeks to utilize WordNet link information must also *discriminate* between them based (at least) on usage statistics of the connected synsets. Also required is an estimate of the likelihood of instantiating a synset into a token because it was "activated" by a closely related synset. We find a Bayesian belief network (BBN) a natural structure to encode such combined knowledge from WordNet and corpus.

2.4 Bayesian Belief Network

A Bayesian Network (Heckerman, 1995) for a set of random variables $X = \{X_1, X_2, \ldots, X_n\}$ consists of a directed acyclic graph (DAG) that encodes a set of conditional independence assertions about variables in X and a set of local probability distributions associated with each variable. Let $\mathbf{Pa}_i$ denote the set of immediate parents of X_i in the DAG, and $\mathbf{pa}_i$ a specific instantiation of these random variables.

The BBN encodes the joint distribution $\Pr(x_1, x_2, \ldots, x_n)$ as

$$\Pr(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} \Pr(x_i|\mathbf{pa}_i) \quad (1)$$

Each node in the DAG encodes $\Pr(x_i|\mathbf{pa}_i)$ as a "conditional probability table" (CPT).

The idea of constructing BBN from WordNet has been proposed by (Rebecca, 1998). But that idea is centered around doing hard-sense disambiguation - to find the 'correct' sense each word in the text.

In this paper, we particularly explore the idea of doing soft sense disambiguation *i.e.* synsets are probabilistically considered to be causes of their constituent words. Moreover, WSD is not an end in itself. The goal is to connect the words within question and answer passage and also across the question and answer passage. WSD is only a by-product.

Our goal is to build a QA system which implements a clear division of labor between the knowledge base and the scoring algorithm, codifies the knowledge base in a uniform manner, and thereby enables a generic algorithm and a shared, extensible knowledge base. Based on the discussion above, our knowledge representation must be probabilistic, and our system must combine and be robust to multiple, noisy sources of information from query and answer terms.

Moreover, we would like to be able to *learn* important properties of our knowledge base from continual *training* of our system with corpus samples as well as samples of successful and unsuccessful (question, answer) pairs. In essence, we would like to automate as far as possible, the customization of lexical networks to QA tasks. Given the English WordNet, it should be possible to reconstruct our algorithm completely from this paper.

Toward these ends, we describe how to induce a Bayesian Belief Network (BBN) from a lexical network of relations. Specifically, we propose a semi-supervised learning mechanism which simultaneously trains the BBN and associates text tokens ,which are words, to synsets in the WordNet in a probabilistic manner ("soft WSD"). Finally, we use the trained BBN to score passages in response to a question.

2.5 Building a BBN from WordNet

Our model of the BBN is that each synset from WordNet is a boolean *event* associated with a question, a passage, or both. Textual tokens are also events. Each event is a node in the BBN. Events can *cause* other events to happen in a probabilistic manner, which is encoded in CPTs. The specific form of CPT we use is the well-known **noisy-OR** of Pearl (Pearl, 1988).

We introduce a node in the BBN for each noun, verb, and adjective synset in WordNet. We also introduce a node for each (non-stop-word) token in the corpus and all questions. Hyponymy, meronymy, and attribute links are introduced from WordNet. *Sense links* are used to attach tokens to potentially matching synsets. E.g., the string "flag" may be attached to synset nodes {sag, droop, swag, flag} and {a conspicuously marked or shaped tail}. (The purpose of probabilistic disambiguation is to estimate the probability that the string "flag" was *caused* by each connected synset node.)

This process creates a hierarchy in which the parent-child relationship is defined by the semantic relations in WordNet. A is a parent of B iff A is the *hypernym* or *holonym* or *attribute-of* or A is a synset containing the word B. The process by which the Bayesian Network is built from the WordNet hypergraph of synsets and and from the mapping between words and synsets is depicted in figure 1. We define *going-up* the hierarchy as the traversal from child to parent.

Ideally, we should update the entire BBN and its CPTs while scanning over the training corpus. In

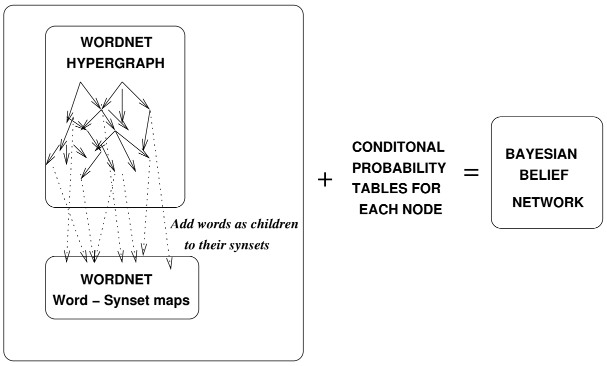

Figure 1: Building a BBN from WordNet and associated text tokens.

practice, BBN training and inference are CPU- and memory-intensive processes.

We compromise by first attaching the token nodes to their synsets and then walking up the WordNet hierarchy up to a maximum height decided purely by CPU and memory limitations. We believe that the probabilistic influence from distant nodes is too feeble and unreliable to warrant modeling.

3 Our QA system

The overall question answering system that we propose is depicted in figure 2.

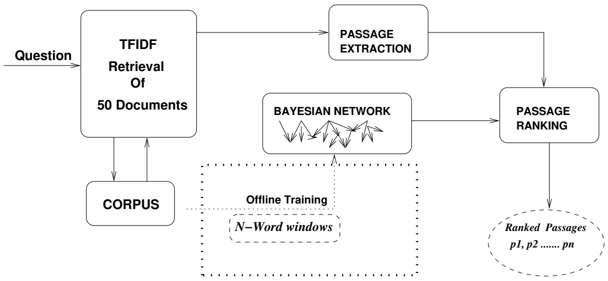

Figure 2: The overall QA system.

The question triggers the TFIDF retrieval module to pick up 50 most relevant documents. These documents are subjected to a sliding window to produce K passages of length N each. The Bayesian belief network described in section 2.5 ranks these passages. The first ranked passage is supposed to contain the answer. The belief network parameters are the CPTs, which are initialized as noisy-or CPTs. The Bayesian belief network is trained offline using the Expectation Maximization algorithm (Dempster, 1977) on windows sliding over the whole corpus.

```
while CPTs do not converge do
    for each window of M words in the text do
        Clamp the word nodes in the Bayesian Network to a state of 'present'
        for each node in Bayesian network do
            find its joint probabilities with all configurations of its parent nodes (E Step)
        end for
    end for
    Update the conditional probability tables for all random variables (M Step)
end while
```

Figure 3: Training the Bayesian Network for a corpus

```
Load the Bayesian Network parameters
for each question q do
    for each candidate passage p do
        clamp the variables (nodes) corresponding to the passage words in network to a state of 'present'
        Find the joint probability of all question words being in state 'present' i.e., Pr(q|p)
    end for
end for
Report the passages in decreasing order of Pr(q|p)
```

Figure 4: Ranking answer passages for given question

3.1 Training the belief network

The figure 3 describes the algorithm for training the BBN obtained from the WordNet. We initialize the CPTs as *noisy-or*. The instances we use for training are windows of length M each from the corpus. Since the corpus is normally not tagged with WordNet senses, all variables, other than the words observed in the window (i.e. the synset nodes in the BBN) are hidden or unobserved. Hence we use the Expectation Maximization algorithm (Dempster, 1977) for parameter learning. For each instance, we find the expected values of the hidden variables, given the present state of each of the observed variables. These expected values are used after each pass through the corpus to update the CPT for each node. The iterations through the corpus are done till the sum of the squares of Kullback-Liebler divergences between CPTs in successive iterations do not differ more than a threshold, or in other words, till the convergence criterion is met. Figure §3 outlines the algorithm for training the Bayesian Network over a corpus. We basically customize the Bayesian Network CPTs to a particular corpus by learning the local CPTs.

3.2 Ranking answer passages

Given a question, we rank the passages with the joint probability of the question words, given the candidate answer. Every question or answer can be looked upon as an event in which the its word nodes are switched to the state 'present'. Therefore, if $p_1, p_2 .... p_n$ are passages and q is the question, the answer is that passage p_i which maximizes $P(q|p_i)$ over all passages p_i deemed as candidate answers. $\Pr(q|p_i)$ is the joint probability of the words of q, each being in state 'present' in the Bayesian network, given that all the word nodes for p_i are clamped to the state 'present' in the belief network. Figure §4 outlines the actual passage ranking algorithm.

The reason for choosing $\Pr(q|p_i)$ over $\Pr(p_i|q)$ is that (a) q typically contains very few words. $\Pr(p_i|q)$, therefore, may not help in bridging the relation between answer words. (b) The passage will be penalized if contains many words which are not present in the question and are also not closely related to the question words through the WordNet. This could happen despite the fact that the passage contains a few words which are all present in the question and/or are semantically closely related to the question, in addtion to containing the answer to the question. Also, (c) if passages p_i's are of varying lengths, $\Pr(q|p_i)$'s are brought to the same scale—that of question words which are fixed across passages/snippets, whereas, $\Pr(p_i|q)$ can be affected and penalized by long snippets.

In fact, our apprehensions about using $\Pr(p_i|q)$ will be justified in the experimental section - the QA performace obtained using $\Pr(p_i|q)$ is drastically poorer - in fact it is worse than the baseline QA algorithm.

Dealing with non-WordNet words: Suppose, there is a word w in the question which is not there in the WordNet. Like the answer passages, we could have ignored such words. But, the question may be seeking an answer to precisely such a word. Also, the number of words being very small in the question, no word in the question should be ignored. We deal with this situation in the following way. We call a word, a *connecting word* if it the key word that links the passage to the question. Note that for WordNet words, the connecting nodes were WordNet concepts. In the case of non-WordNet words, we don't have any hidden, connecting nodes. So we consider the words themselves to be possible connections.

Let $connectw$ be a random variable which takes the state 'present' if w is a connecting word between the question and the answer. It's state is 'absent' if it is not a connecting word. Let wq, wp be random variables that are 'present' if w occurs in the question or answer respectively, else they are 'absent'.

By Bayes rule, we get the following probability that the word w occurs in the question, given that it occurs in the answer (1=Present, 0=absent).

$$\Pr(wq = 1|wp = 1) \approx$$

$$\Pr(wq = 1|connectw = 1) \times \Pr(wp = 1|connectw = 1) \times$$

$$\Pr(connectw = 1) +$$

$$\Pr(wq = 0|connectw = 0) \times \Pr(wp = 0|connectw = 0) \times$$

$$\Pr(connectw = 0)$$

where $Pr(connectw = 1)$, $Pr(wq = 1|connectw = 1)$, $Pr(wp = 1|connectw = 1)$, and $Pr(connectw = 1)$ and their complements are estimated from question answer pairs. Moreover, the occurrence of non WordNet words is assumed to be independent of each other and also of the occurrence of WordNet words.

4 Use of Regular Expressions for passage Filtering

A study of the available question-answer pairs from the earlier TREC releases, helped us to identify patterns for filtering passages corresponding to every question type. The question type is identified for a group of question cue phrases. For every group, a regular expression is identified. A question cue phrase can belong to more than one group of cue phrases.

For example, the group of cue phrases that belong to the class of DURATION such as how_long how_often, how_short, how_frequently, how_far how_fast, how_swift, how_old and how_new make it manditory for the answer to contain regular expressions such as $[CD]+$. The regular expressions that were used were quite involved. A balance between general versus specific regular expression needs to be achieved since very general regular expressions do not serve any purpose in the answer phrase filtering while very specific regular expressions give a very low recall.

5 BBN simplification

Following are some of our observations regarding the approach of Bayesian Inferencing for identifying the answer passages.

5.1 Observation on variety of dependency arcs in BBN

In the preliminary experiments with Bayesian Inferencing, we initialized all CPTs as noisy−or. Noisy−or CPTs make sense for nodes which could be caused exclusively by one of the parents *i.e.* when the occurrence of one parent event precludes the occurrence of other parent events. This is true for word nodes, whose parents are factually its different senses and occurrence of one sense of a word, precludes occurrence of its other senses.

But this is not true for nodes that are synsets − the parents of such nodes are compositional in nature − the child is simultaneously the hyponym of some of its parents and the meronym of its other parents. Hence, we need to revamp our approach of using a noisy−or model for the entire network, from the leaf nodes corresponding to the words upto the root nodes. We propose to initialize the words nodes with noisy−or CPTs and synset nodes with noisy−and CPTs.

5.2 Observation on senses of a word

Our observation is that word senses as given by WordNet, or for that matter word senses given by any lexicon, are not completely orthogonal or unrelated. In fact, to different extents, word senses overlap and form soft−clusters. We feel that any algorithm that attempts to exploit relations between word senses must explicitly take care of this fact. In doing bayesian inferencing with the whole network, we tried to capture this phenomenon implicitly through the idea of *soft sense disambiguation*. But this was at the cost of computationally and memory intensive algorithms. We are working in the direction of simplifying the network before−hand. We present some observations and a simple algorithm in pursuit of the goal.

The observation is that word−senses with similar ancestral lineage have more overlap in their meanings than those with completely distinct ancestral lineages.

6 Discussion and future work

We have described a passage-scoring algorithm for QA via Bayesian inference on lexical relations. By separating the inference algorithm from the design of the knowledge base, we made our system extensible and trainable from a corpus.

Our work hinges upon the existence of lexical relations in the WordNet. We would like to point out here that no special efforts were made in the construction of the Bayesian Network from WordNet nor did we attempt to fill in the desirable 'missing links' between words or synsets in WordNet or remove spurious links in WordNet. Thus, we are able to find probabilities based on semantic relations to the extent given by links in WordNet and we are able to uncorrelated words from each other to the extent they are disconnected in WordNet. To some extent, we attempt to learn the Bayesian Network parameters and this does result in improvement in Question Answering performance. But it will be interesting to see if training the network with bigger corpora improves the performance further. Another experiment that remains to be tried is training the Bayesian Network with samples of successful and unsuccessful (question, answer) pairs.

One thing to note is that if all the question words are contained in the passage, the passage will get a high rank because it will induce a joint probability score of 1 on the question. This can happen even if the answer is not contained in the passage.

Another limitation is the computational and memory cost. On an average it took 0.03 seconds for Bayesian inferencing on a passage. The memory requirement goes upto 30MB. One future work will comprise of reducing the online memory and computational requirements by simplifying the network structure and/or making certain computations offline.

We would also like to find better initial values to speed up learning and avoid local optima. We would like to re-introduce the notion of lexical proximity into our inference process, so as to further improve the accuracy of WSD. We also wish to explore how continual feedback and retraining of the BBN can improve the accuracy of our system.

References

Abe, Naoki, and Hang Li. 1996. Learning word association norms using tree cut pair models. In *Proceedings of the 13th International Conference on Machine Learning*.

C. Buckley. 1985. Implementation of the smart information retrieval system. Technical report, Technical Report TR85-686, Department of Computer Science, Cornell University.

C. L. A. Clarke, Gordon V. Cormack, and Thomas R. Lynam. 2001. Exploiting redundancy in question answering. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 358–365. ACM Press.

C. Fellbaum, 1998. *WordNet: An Electronic Lexical Database*, chapter Using WordNet for Text Retrieval, pages 285–303. The MIT Press: Cambridge, MA.

Christiane Fellbaum. 1998b. *WordNet: An Electronic Lexical Database*. The MIT Press.

Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu. 2000. Falcon: Boosting knowledge for answer engines. In *Proceedings of the ninth text retrieval conference (TREC-9)*, November.

David Heckerman. 1995. A Tutorial on Learning Bayesian Networks. Technical Report MSR-TR-95-06, March.

Boris Katz. 1997. From sentence processing to information access on the world wide web. *AAAI Spring Symposium on Natural Language Processing for the World Wide Web, Stanford University, Stanford CA*.

Cody C. T. Kwok, Oren Etzioni, and Daniel S. Weld. 2001. Scaling question answering to the web. In *Proceedings of the Tenth International World Wide Web Conference*, pages 150–161.

David D. Lewis and Karen Sparck Jones. 1996. Natural language processing for information retrieval. *Communications of the ACM*, 39(1):92–101.

J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc.

Adwait Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference, May 17-18, 1996. University of Pennsylvania*.

Mark Sanderson. 1994. Word sense disambiguation and information retrieval. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 49–57, Dublin, IE.

Ellen Vorhees. 2000. Overview of TREC-9 question answering track. *Text REtreival Conference 9*.

Wiebe, Janyce, O'Hara, Tom, Rebecca Bruce. 1998. Constructing Bayesian networks from WordNet for word sense disambiguation: representation and processing issues. In *Proc. COLING-ACL '98 Workshop on the Usage of WordNet in Natural Language Processing Systems*.

.P. Dempster, N.M. Laird and D.B. Rubin. 1977. Maximum Likelihood from Incomplete Data via The EM Algorithm. In *Journal of Royal Statistical Society*, Vol. 39, pp. 1-38, 1977.

Ganesh Ramakrishnan and Pushpak Bhattacharyya. 2003. Text Representation with WordNet Synsets: A Soft Sense Disambiguation Approach. To appear in *Proceedings of the* 8^{th} *International Conference on Natural Language in Information Systems*, Springer Verlag.