Question answering via Bayesian inference on lexical relations

Ganesh Ramakrishnan, Apurva Jadhav, Ashutosh Joshi, Soumen Chakrabarti, Pushpak Bhattacharyya
{hare,apurvaj,ashuj,soumen,pb}@cse.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. Our system is superior to vector-space ranking techniques from IR, and its accuracy approaches that of the top contenders at the TREC QA tasks in recent years.

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).

1.1 Question answering (QA)

Unlike IR systems which return a list of documents in response to a query, from which the user must extract the answer manually, the goal of QA is to extract from the corpus direct answers to questions posed in a natural language.

An important step before answer extraction is to identify and rate candidate *passages* from documents which might contain the answer. The notion of a passage is somewhat arbitrary: various notions of a passage have emerged (Vorhees, 2000); For our purposes, a passage comprises M consecutive sentences, or N consecutive words.

In contrast to IR, where linguistic resources have not been found very useful, QA has always depended on a mixture of stock lexical networks and custom ontologies (language-independent conceptual hierarchies) crafted through human understanding of the task at hand (Harabagiu et al., 2000; Clarke et al., 2001). Ontologies, hand-crafted and customized, sometimes from the WordNet itself, are employed for question type classification, relationships between places, measures, *etc.*

The scoring (and thereby, ranking) of passages through lexical networks or ontologies is more successful in QA than in classic IR because of the nature of the QA task. Passage-scoring in QA benefits from indirect matches through an ontology.

By separating the passage-scoring algorithm from the knowledge base, we can keep improving our system by continually upgrading the lexical relations in the knowledge base and retraining our inference algorithm.

Map: §2 describes the related work. §3 gives the motivation behind our approach and the background information (WordNet and Bayesian inferencing). §4 describes our QA system. Results are presented in §5, and concluding remarks made in §6.

2 Related work

Information Retrieval (IR) systems such as SMART (Buckley, 1985) rank documents for relevance w.r.t. to a user query, based on keyword match between the query and a document, each represented in the well-known "vector space model". The degree of match is measured as the cosine of the angle between query and document vectors.

In QA, an IR subsystem is typically used to short-list passages which are likely to embed the answer. Usually, several enhancements are made to stock IR systems to meet this task.

First, the cosine measure used in stock vector-space systems will be *biased against long documents* even if they embed the answer in a narrow zone. This problem can be ameliorated by representing suitably-sized passage windows (rather than whole documents) as vectors. While scoring passages using the cosine measure, we can also ignore passage terms which do not occur in the query.

The second issue is one of *proximity*. A passage is likely to be promising if query words occur close to one another. Commercial search engines reward proximity of matched query terms, but in undocumented ways. Clarke *et al.* (Clarke et al., 2001) exploit term proximity within documents for passage scoring.

The third and most important limitation of stock IR systems is the inability to *bridge the lexical chasm* between question and potential answer via lexical networks. One query from TREC (Vorhees, 2000) asks, "Who painted Olympia?" The answer is in the passage: "Manet, who, after all, created Olympia, gets no credit."

QA systems use a gamut of techniques to deal with this problem. FALCON (Harabagiu et al., 2000) (one of the best QA systems in recent TREC competitions) integrates syntactic, semantic and pragmatic knowledge for QA. It uses WordNet-based query expansion to try to bridge the lexical chasm. WordNet is customized into a answer-type taxonomy to infer the expected answer type for a question. Named-entity recognition techniques are also employed to improve quality of passages retrieved. The answers are finally filtered by justifying them using abductive reasoning. Mulder (Kwok et al., 2001) uses a similar approach to perform QA on Web scale. The well-known START system (Katz,) goes even further in this direction.

Discussion: In general, the TREC QA systems divide QA into two tasks: identifying relevant documents and extracting answer passages from them. For the former task, most systems use traditional IR engines coupled with ad-hoc query expansion based on WordNet. Handcrafted knowledge bases, question/answer type classifiers and a variety of heuristics are used for the latter task. Success in QA comes at the cost of great effort in custom-designed wordnets and ontologies, and expansion, matching and scoring heuristics which need to be upgraded as the knowledge bases are enhanced. Ideally, we should use a knowledge base which can be readily extended, and a core scoring algorithm which is elegant and "universal".

3 Proposed approach

3.1 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. This scenario is depicted in figure §1. The causes themselves may have hidden causes associated with them.

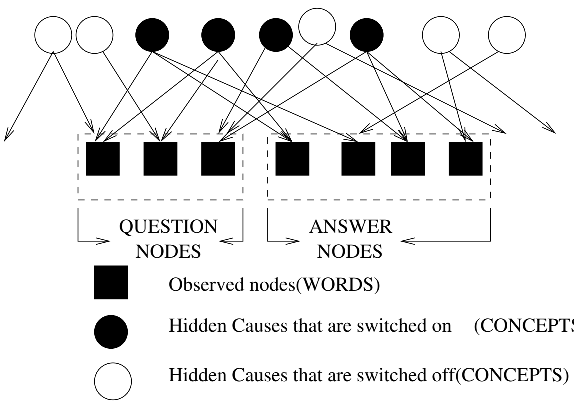

Figure 1: Motivation

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}.

3.2 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.

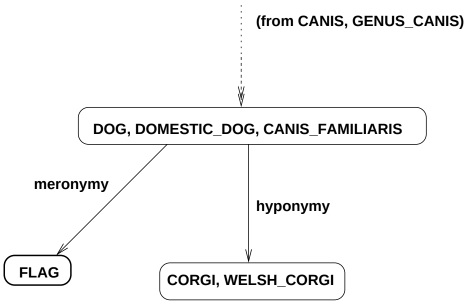

Figure 2: Illustration of WordNet relations.

Figure§2 shows that 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.

3.3 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 §3.1), 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.

3.4 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

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

Each node in the DAG encodes $\Pr(x_i|\mathbf{pa}_i)$ as a "conditional probability table" (CPT). Figure §3 shows a Bayesian belief network interpretation for a part of WordNet. The synset {*corgi, welsh_corgi*} has a causal relation from {*dog, domestic_dog, canis_familiaris*}. A possible conditional probability table for the network is shown to the right of the structure.

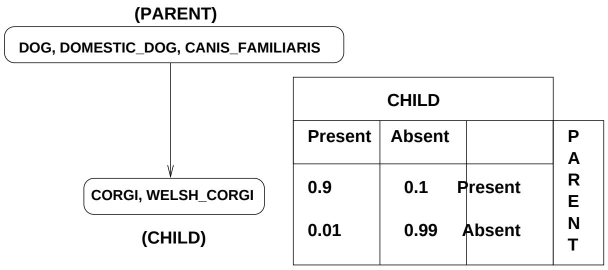

Figure 3: Causal relations between two synsets.

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.

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

We compromise by first attaching the token nodes

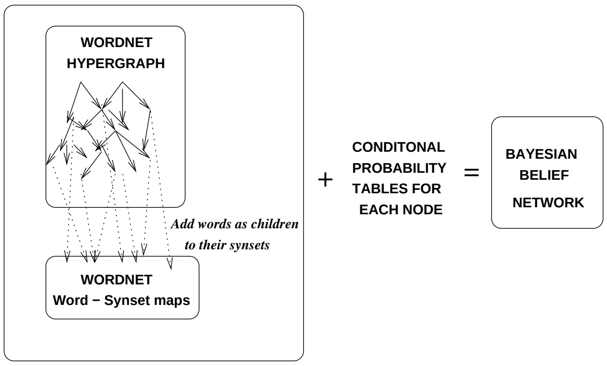

Figure 4: Building a BBN from WordNet and associated text tokens.

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.

4 Our QA system

The overall question answering system that we propose is depicted in figure 5. The corresponding algorithm is outlined in figure 6.

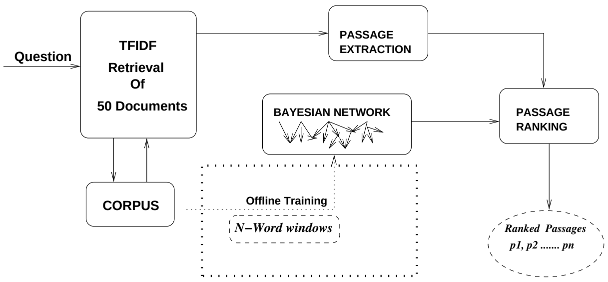

Figure 5: The overall QA system.

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 3.5 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

1: Construct a Bayesian Network structure using the WordNet structure
2: Train the Bayesian network parameters on the corpus containing the answers
3: Do question answering with trained Bayesian Network

Figure 6: The over-all question answering algorithm

```
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 7: Training the Bayesian Network for a corpus

the Expectation Maximization algorithm (Dempster, 1977) on windows sliding over the whole corpus.

4.1 Training the belief network

The figure 7 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 §7 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.

4.2 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.

1: Load the Bayesian Network parameters
2: **for** each question q **do**
3: **for** each candidate passage p **do**
4: clamp the variables (nodes) corresponding to the passage words in network to a state of 'present'
5: Find the joint probability of all question words being in state 'present' i.e., $\Pr(q|p)$
6: **end for**
7: **end for**
8: Report the passages in decreasing order of $\Pr(q|p)$

Figure 8: Ranking answer passages for given question

Figure §8 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.

5 Experiments and results

We perform extensive experiments to evaluate our system, using the TREC `http://trec.nist.gov/data/qa.html` QA benchmark. We find that our algorithm is a substantial improvement beyond a baseline IR approach to passage ranking. Based on published numbers, it also appears to be in the same league as the top performers at recent TREC QA events. We also note that training our system improves the quality of our ranking, even though WSD accuracy does not increase, which affirms the belief that passage scoring need not depend on perfect WSD, given we use a robust, 'soft WSD'. See section §3.3.

5.1 Experimental setup

We use the Text REtrieval Conference (TREC) (Vorhees, 2000) corpus and question/answers from its QA track. The corpus is 2 GB of newspaper articles. There is a set Q of about 690 factual questions. For each question, we retrieve the top 50 documents using a standard TFIDF-based IR engine such as SMART. We used the question set and corresponding top 50 document collection from TREC 2001 for our experiments. We used MXPOST (Ratnaparkhi, 1996), a maximum entropy based POS tagger. The part of speech tag is used while mapping document and question terms to their corresponding nodes in the BBN.

The passage length we chose was $N = 20$ words. Unless otherwise stated explicitly, the maximum height upto which the BBN was used for inferencing for each Q-passage pair can be assumed to be 4.

5.2 Evaluation

TREC QA evaluation has two runs based on the length of system response to a question. In the first the response is a passage up to 250 bytes in size. The second, more ambitious run asks for shorter responses of up to 50 bytes. (More recently, TREC has updated its requirements to demand exact, extracted answers.)

To determine if the response is actually an answer to the question, TREC provides a set of regular expressions for each question. The presence of any of these in the response indicates that it is a valid answer. For evaluation the system is required to submit its top five responses for each question. This is used to calculate the performance measure **mean reciprocal rank** (MRR) for the system, defined as

$$\mathrm{MRR} = \frac{1}{|Q|}\left(\sum_{q\in Q}\frac{1}{\mathrm{rank}_q}\right). \qquad (2)$$

Here rank_q is the first rank at which correct answer occurs for question $q \in Q$. If for a question q the correct answer is not in the top 5 responses then $\frac{1}{\mathrm{rank}_q}$ is taken to be zero.

5.3 Results

IR baseline: IR technology is widely accessible, and forms our baseline. We construct 250-byte windows of text as passages and compute the similarity between these passages and the query. Because we would not like to penalize passages for having terms not in the question (provided they have at least some query terms), we use an asymmetric TFIDF similarity. Under this measure, the score of a passage is the sum of the IDFs of the question terms contained in the passage. If D is the document collection and D_t is the set of documents containing t, then one common form of IDF weighting (used by SMART again) is

$$\mathrm{IDF}(t) = \log\frac{1+|D|}{|D_t|}. \qquad (3)$$

The IR baseline MRR is only about 0.3, which is far short of Falcon, which has an MRR of almost 0.7. The baseline MRR is low for the obvious reasons: the IR engine cannot bridge the lexical gap.

System	**MRR**
Asymmetric TFIDF	0.314
Untrained BBN	0.429
Trained BBN	0.467

Table 1: MRRs for baseline, untrained and trained BBNs

System	MRR
FALCON	0.76
University of Waterloo	0.46
Queens College, CUNY	0.46

Table 2: MRRs for best performing systems in TREC9

Base BBN: Initialized with our default parameters, our BBN-based approach achieves an MRR of 0.429, which is already a significant step up from the IR baseline. A large component of this improvement is caused by conflating different strings to common synsets.

Trained BBN: We recalibrated our system after training the BBN with the corpus. This resulted in a visible improvement in our MRR, from 0.429 to 0.467, which takes us into the same league as the systems from University of Waterloo and Queens College, reported at TREC QA.

Tables §1 and §2 summarize our MRR results and juxtapose them with the published MRRs for some of the best-performing QA system in TREC 2000. Given that we have invested **zero** customization effort in WordNet, it is impressive that our MRR compares favorably with all but the best system.

Experiments for varying heights of BBN: The MRR obtained went down to 0.34 when the height of the traced BBN was restricted to 1, *i.e.* only words and their immediate synsets were considered. It is significant to note that even with immediate synset expansion, there is a marginal improvement over asymmetric TFIDF. The MRR improved to 0.42 and 0.45 when the height was increased to 2 and 3 respectively. These results are tabulated in table §3.

Experiments for restricting to WordNets of different parts of speech: The MRR found by using only the noun WordNet was 0.415. Words in the remaining parts of speech were treated as

Height	**MRR**
1	0.342
2	0.421
3	0.450
4	0.467

Table 3: MRRs for BBNs truncated at different heights

WordNet for diff POS	MRR
Noun	0.415
Adjective	0.340
Verb	0.32
Noun+Adjective	0.442
Noun+Verb	0.393
Verb+Adjective	0.332
Noun+Verb+Adjective	0.467

Table 4: MRRs for BBNs restricted to diff parts of WordNet

Expt setup	MRR
$P(Q\|A)$ with only WNet words	0.370
No Bayesian Inferencing	0.30
$P(Psg\|Question)$	0.021

Table 5: MRRs for other experiments

non WordNet words in this experiment. The MRR dropped to 0.340 when only the adjective WordNet was used. The MRR found using only the verb WordNet was a low 0.32. This is because the verb WordNet is very shallow and many semantically distant verbs are connected closely together. The MRR score obtained by considering noun+adjective part of WordNet was 0.442, that obtained by considering noun+verb part of WordNet was 0.393 and that obtained by considering verb+adjective part of WordNet was 0.333. These results are summarized in table §4. The results seem to justify the observation that the verb WordNet in its current form is shallow in height and has high in/out degree for each node; this is mainly due to the high ambiguity of verbs. But coupled with noun and adjective WordNets, the verb WordNet improves overall performance.

Miscellaneous experiments: The MRR obtained by considering only WordNet words was 0.370 which indicates that we cannot afford to ignore the non-WordNet words. Also it seems that inducing 'semantic-similarity' between words not in the WordNet vocabulary is not so much required. By skipping Bayesian inferencing altogether, we get an MRR of 0.30 which is the same as for asymmetric TFIDF mentioned earlier. The MRR drastically fell to 0.021 when $P(Psg|Question)$ was used to rank the passages. This partly justifies the apprehension about finding the probability of passage given question which was expressed earlier - that is, passages get penalized if they contain lots of words which are not either not there in the question or are not related to words in the question. These results are summarized in table §5.

The effect of WSD: It is interesting to note that training does not substantially affect disambiguation accuracy (which stays at about 75%), and MRR improves *despite* this fact. This seems to indicate that learning joint distributions between query and candidate answer keywords (via synset nodes, which are "bottleneck" variables in BBN parlance) is as important for QA as is WSD. Furthermore, we conjecture that "soft" WSD is key to maintaining QA MRR in the face of modest WSD accuracy.

5.4 Analysis

In the following, we analyse how Bayesian inferencing on lexical relations contributes towards ranking passages.

How joint probability helps For finding the probability of question given a passage, we take the joint probability of the question words, conditioned on the (evidence of) answer words. Thus we attempt to overcome the usual bottleneck of assumption of independence of words as in the naive Bayes model. The relations of question words between themselves and with words in the answer is what precisely helps in giving a joint probability that is different from a naive product of marginals. This will be illustrated in section §5.5.

How parameter smoothing helps If a question word does not occur in the answer, the marginal probability of that word should be high if it strongly relates to one or more words in the answer through WordNet. Without using WordNet, one could resort to finding this marginal probability from a corpus. These probabilities are remarkably low even for words that are very semantically related to words in the answer and this will be illustrated in the case studies in section §5.5. This problem could be attributed to data sparsity

5.5 Case studies

Case 1: This example shows that the passage in figure §10 contains the correct answer to the question in figure §9 and was given rank 1. The interesting observation is that the words *kind* and *type* are related correctly through the WordNet to give high marginal probability to the word *kind* (0.557435) in the question, even though it does not occur in the answer. This is depicted in figure §12.

The marginal probability of the same word (given that its is absent in the answer passage), as determined by corpus statistics is 0.00020202 - which is very small. This illustrates the advantage of parameter smoothing.

TREC Question ID 371: A corgi is a kind of what?

Figure 9: Sample question Q1

Bayesian Marginal Probs: corgi: 1.000000, kind: 0.557435 ....corgis: They are of course collie-type **dogs** originally bred for cattle herding. As such they will chase anything particularly ankles....

Figure 10: Answer for Q1, Rank 1, Score(Joint Probability) = 0.893133, (Document ID:AP881106-0015)

Bayesian Marginal Probs: corgi:1.000000, kind:0.006421 ....current favorite. So are bulldogs. Jack Russell terriers are popular with the horsy set. “ The short-legged welsh_corgi is big (QueenP elizabeth_ii has at least one). And so, of course, is the_english bull_terrier (thanks to Anheuser-Busch, Bud Light and Spuds. MacKenzie). Barbara.....

Figure 11: Non-answer for Q1, Rank 2, Score(Joint Probability) = 0.647734, (Document ID:WSJ900423-0005)

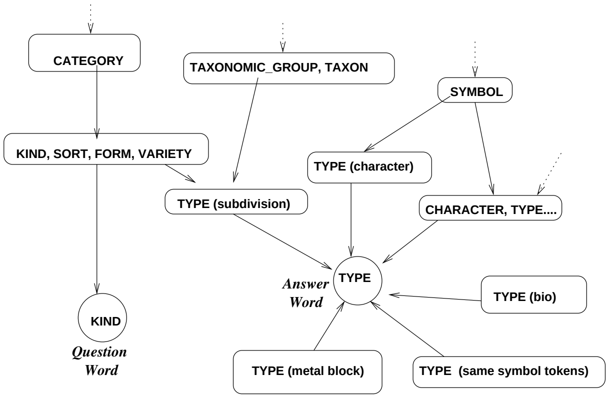

Figure 12: Relation between *kind* in question and *type* in answer

Additionally, the joint probability of question words given the passage words of figure §10 (0.893133) is not the product of their marginals ($P(corgi|PASSAGE) = 1.000000$, $P(kind|PASSAGE) = 0.557435$). The reason for this is that the word *dog* that occurs in the answer passage is related to the word *corgie* in the question through WordNet as shown in figure§13. It can be seen easily that these lexical relations increase the joint probability of the question words, given the answer words, over the product of the marginals of the individual words.

In contrast, the passage of figure §11 which contains no answer to the question, also contains no word which is closely related to the word

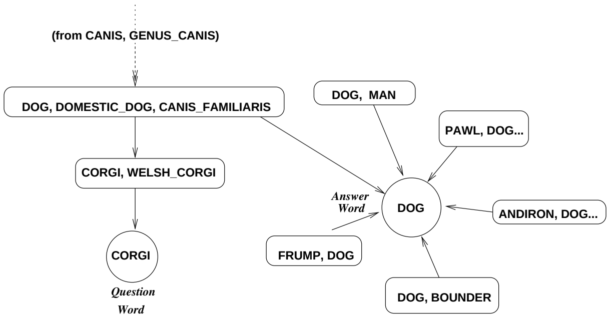

Figure 13: Relation between *dog* in answer and *corgi* in question

kind through WordNet. Therefore, the marginal probabilities as well as the joint probability of same question words given this passage are low as compared to the passage of figure§10. As a result the second passage gets a low rank.

Case 2: The passage in figure §15 was highest ranked for the question in figure§14, even though it does not contain the answer *central america*. This is because, all question words occur in the passage and therefore, the passage gets a rank of 1. This highlights a limitation of our mechanism. On the other hand, the passage ranked 2^{nd} contains the answer. It gets a joint probability score of 0.890192, even though the word *belize* does not occur in the answer. This is because *belize* is connected to the word *central america* and also to *country* through WordNet. The passage shown in figure§17, which does not contain the answer, got a pretty low rank of 10 because it induced a low joint probability of 0.033451 on the question even though the word *belize* was present in the passage, because *locate* was absent in the passage and it is not immediatly connected to other words in the passage. This again illustrates the advantage of using Bayesian inferencing on lexical relations.

Case 3: Here we present an example to illustrate where the mechanism can go wrong due to the of absence of links. The passage in figure §19 induces a conditional joint probability of 1 on the question in figure §18, because the passage contains all the words present in the question. The passage however does not answer the question. On the other hand, the passage shown in figure §20 contains the answer, but induces a lower joint probability on the question - because the verb *stand_for* is not closely related, through WordNet to any of the words in the passage. In fact, one would have expected *stands_for* and *stand_for* to be related to each other through

TREC Question ID : 202 Where is Belize located ?

Figure 14: Sample question Q2

Bayesian Marginal Probs: belize: 1.000000, locate: 1.000000 ....settlers has been confirmed to the east of the historic monuments that are being used as a reference_point with Belize . She pointed out that in case they prove the settlement is located in the protected Mayan biosphere area and that it was established illegally , the settlers will have to leave the area , but the.....

Figure 15: Non-Answer to Q2, Rank = 1, Score(Joint Probability) = 1, DocID: FBIS3-10202

Bayesian Marginal Probs: belize: 0.889529, locate: 1.000000 ....confirmed that the Belizean Government will assume responsibility for its own defense as of 1 January 1994 and announced that it had started the " immediate withdrawal of the UK troops stationed in that country located in the **central american** isthmus . Lourdes ......

Figure 16: Answer to Q2, Rank = 2, Score(Joint Probability) = 0.890192, DocID: FBIS3-50428

Bayesian Marginal Probs: belize: 1.000000, locate: 0.033451 ....prepared to begin negotiations on the territorial dispute with Guatemala ;_: adding that a commission has been created for this purpose and only the final details must be settled . The Guatemalan Government has recognized Belize 's independence ;_: therefore , we have accepted the fact that a .....

Figure 17: Non-answer for Q2, Rank = 10, Score(Joint Probability) = 0.452310, DocID: FBIS4-56830

TREC Question ID : 224 What does laser stand_for ?

Figure 18: Sample question Q3

Bayesian marginal Probs: laser: 1.000000, stand_for: 1.000000 ....Yu.A. Rezunkov , candidate of technical sciences , department_head , V.S. Sirazetdinov , candidate of technical sciences , manager of test stand_for adaptive laser systems , A.V . Charukhchev ,.....

Figure 19: Non-Answer to Q3, Rank = 1, Score(Joint Probability) = 1, DocID: FBIS4-47304

Bayesian marginal Probs: laser: 1.000000, stand_for: 0.073516 ...Laser stands for **light amplification by stimulated emission of radiation**. Both masers and lasers are devices containing crystal , gas or other substances that get atoms so excited as they bounce_back and forth in step between two mirrors that they finally burst_out in one coherent .....

Figure 20: Answer to Q3, Rank = 25, Score(Joint Probability) = 0.890192, DocID: FBIS3-50428

Bayesian marginal Probs: laser: 1.000000, stand_for: 0.060797
Corpus based-marginal Probs: laser: 0.990561, stand_for: 2.886e-05 ....surface plasma by interaction of laser radiation and solid targets covering the 10 [.sup] 5 [/] -10 [.sup] 10 [/]_. range of radiation intensity being essentially considered here along with negative and positive.....

Figure 21: Non-answer for Q3, Rank = 50, Score(Joint Probability) = 0.86329, DocID: FBIS4-22835

WordNet.

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. The accuracy of our system is better than IR-like scoring techniques, and compares favorably with well-known QA systems, as shown in section 5.

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.