A Computational Cognitive Model for Semantic Sub-network Extraction from Natural Language Queries

Suman DEB ROY   Wenjun ZENG

Department of Computer Science
University of Missouri – Columbia, USA.

sdr5x8@mail.missouri.edu, zengw@missouri.edu

ABSTRACT

Semantic query sub-network is the representation of a natural language query as a graph of semantically connected words. Such sub-networks can be identified as sub-graphs in larger ontologies like DBpedia or Google knowledge graph, which allows for domain and concepts identification, especially in noisy queries. In this paper, we present a novel standalone NLP technique that leverages the cognitive psychology notion of semantic *forms* for semantic sub-network extraction from natural language queries. Semantic *forms*, borrowed from cognitive psychology models, are one of the fundamental structures employed by human cognition to construct semantic information in the brain. We propose a computational cognitive model by means of conditional random fields and explore the interaction patterns among such *forms*. Our results suggest that the cognitive abstraction provided by semantic *forms* during labelling can significantly improve parsing and sub-network extraction compared to pure lexical approaches like parts of speech tagging. We conduct experiments on approximately 5000 queries from three diverse datasets to demonstrate the robustness and efficiency of the proposed approach.

KEYWORDS: Cognitive Linguistics, Forms, Query, Search, Subnets, Semantic

Proceedings of COLING 2012: Technical Papers, pages 727–744,
COLING 2012, Mumbai, December 2012.

1 Introduction

The efficiency of natural language (NL) search often depends on detection of keywords in a query, followed by construction of some meaningful connected network comprising of such keywords (Herdagdelen, 2010). This connected network of keywords is called a semantic subnet (Booth, 2009). These keywords together comprise what is called a semantic field (Croft, 2003). The goal of our research is to efficiently recover the semantic sub-network from NL queries.

Prior research suggests three main motivations for extracting semantic subnets from NL queries. Firstly, extracted query subnets can be used to generate a candidate set of concepts within a larger ontology (like of DBpedia RDF network/ Google knowledge graph), which may align to the words in the query subnet and assist domain identification and query expansion (Booth, 2009). Secondly, a query subnet can act as a NL interface to concept graph databases (Popescu, 2003), facilitating semantic information retrieval (Kauffman, 2007) and improved query understanding and semantic search (Hu, 2009). Finally, semantic subnets enable identification of event structures within sentences (McClosky, 2011) and assist higher-level NLP tasks, like Question Answering (QA) (Huang, 2009).

It is possible to detect semantic keywords in NL queries using methods like Named Entity Recognition (NER) or Semantic Role Labelling (SRL) (Collobert, 2011). Both techniques provide a higher level of abstraction than the basic syntax tree. However, our task goes a step further: we aim to find out *how these keywords are semantically connected* in terms of a network. This is very difficult to achieve using NER alone, since detecting the named entities provides limited information about their relations. SRL does a better job at concept level parsing using predicate logic, but is bound by the strict predicate grammar. Therefore, although techniques such as NER and SRL is core to NLP, there is an inherent gap between requirements of intelligent tasks (like QA) and several state-of-the-art NLP techniques (Finkel, 2009).

As search is becoming more collaborative and social, queries turn noisier (Hu, 2009). Often, the conceptual structure of the NL query is difficult to extract using simple (Parts-Of-Speech) POS-based dependency parsing. Imprecision of NL usage is a major obstacle to computation with NL. Therefore, it is necessary to develop a technique that partially relaxes the rigid grammar of the language. While imprecise or varied grammatical constructions are difficult to capture using POS or predicate logic, note that the human cognition can often eliminate such noise to interpret meaning. If we assume that 'meaning' of a NL sentence is captured in its semantic subnet, then it would be logical to conclude that human cognition possesses a more noise-resistant process of extracting semantic subnets. A rational explanation for this is the presence of an improved model for detecting semantics in NL and subsequently constructing semantic information in the brain.

Cognitive psychology has a number of interesting theories on how the human mind deals with imprecision, uncertainty and complexity of language (Chater, 2006). One such theory, called the structure-of-intellect model, proposes that humans perceive concepts contained within the words of a sentence as a semantic *form* (Guilford, 1977). His model has been widely used to study the cognitive intellect and the kinds of information that humans can extract from any observed argues that human cognition is robust to noise because it dynamically changes the resolution at which data is to be semantically interpreted (Croft, 2004).

Recognizing the potential of cognitive approaches in semantic information modelling, we propose to leverage semantic *forms* in the extraction of semantic sub-networks from NL queries. These semantic *forms*, when connected in some networked pattern, becomes responsible for understanding the scope and context of a concept, and assists functional retrieval of related concepts and question answering/response (Carroll, 1993). Thus, our main insight in modelling semantic *forms* and their interaction patterns in NL is grounded on the idea: *the subsurface form space demonstrates the query intent (expresses semantics) better than superficial (lower) query syntactical features, which might vary depending on diverse query construction.* In other words, the higher is the level of abstraction for labelling, the more robust the extraction should become. This idea of cognitive abstraction provided by semantic *forms* is shown in Fig. 1.

The main contributions of this paper are:

- We propose the use of semantic *forms*, borrowed from cognitive science, as label category for NL sequence labelling tasks.
- We propose a conditional random field based method of implementing the structure of intellect model, by labelling query words with semantic *forms* and analyzing the interconnected patterns in which such *forms* exist within a semantic field.
- We perform experiments on three diverse query datasets consisting of TREC, QA-type and Web queries to justify the robustness of our approach to varying noise levels. Our approach comprehensively outperforms existing works on query subnet detection (Booth, 2009).

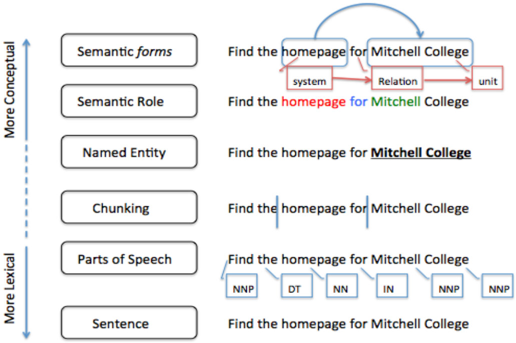

More Conceptual
More Lexical
Semantic *forms*
Semantic Role
Named Entity
Chunking
Parts of Speech
Sentence
Find the homepage for Mitchell College
system
Relation
unit
Find the homepage for Mitchell College
Find the homepage for Mitchell College
Find the homepage for Mitchell College
Find the homepage for Mitchell College
NNP
DT
NN
IN
NNP
NNP
Find the homepage for Mitchell College

FIGURE 1 – Level of abstraction in different NLP techniques: from lexical to conceptual.

The rest of this paper is organized as follows: in Section 2 we discuss related work and the scope of the paper. Section 3 introduces the notion of semantic *forms* and their interactions. In Section 4, we describe the proposed model for labelling query words with linked semantic *forms*. Section aids better NL interfaces that precisely map a NL query into a suitable graph database. In the next two sub-sections, we describe the related work and scope of this paper.

2.1 Related Work

Semantic subnet extraction from NL queries is essentially a method of query reformulation, wherein a query is represented in an alternative form that eases its interface with different types of databases for concept detection. A detailed analysis of query reformulation techniques is available in (Clifford, 1990) and (Herdagdelen, 2010). Substantial efforts have been exhausted in trying to enhance the role of NL interfaces in converting a natural language query into a graph database query (Booth, 2009) (Popescu, 2003). A semantic network of Resource Description Format (RDF) concepts (DBpedia) can be considered one such graph database (Auer, 2007). Several problems like word-sense disambiguation (Bruce, 1994), specificity of grammar (Manning, 1999) and keyword (not semantic) based approaches inhibit portability of several existing NLP techniques across systems and domains (Kaufmann, 2007). The closest work to our research is (Booth, 2009), which uses a POS-based approach in extracting subnets from queries. The accuracy of query subnet extraction compared to a human standard can be evaluated using metrics such as Consistency Index (Cardona, 2009). The results stated in (Booth, 2009) are tested on a very limited number of queries (approx. 12), which does not come close to capturing the diversity in human query constructions or web scale. In contrast, we provide empirical results on 5000 queries from three query datasets with different noise levels.

Substantial efforts have also been spent in detecting named entities in sentences. This task, called Named Entity Recognition (NER) seeks to locate and classify words such as names, organizations, places etc. in a NL sentence (Guo, 2009). A similar NLP task is SRL, which involves finding target verbs (predicate) in a sentence that resemble some ‘action’ (Collobert, 2011). Models have also strived to combine these two techniques (Finkel, 2009). In this paper, we will use Conditional Random Fields (CRF) (Lafferty, 2001) to capture the cognitive model in terms of finite state automata, with each *form* state dependent on the previous *form* state and on the current symbol word being processed. The resulting Markov chain of possible states can be solved using the Viterbi Algorithm, implemented using dynamic programming (Manning, 1999).

J. P. Guilford introduced the structure-of-intellect model in (Guilford, 1977), which covers the notion of semantic *forms* as ‘products’. ‘Products’ are the result of applying some cognitive operation (cognition, retention etc.) on specific content (semantic, symbolic etc.). The model has since been used, studied and analysed substantially in the cognitive science community. A detailed view of human cognitive semantics in linguistics is provided in (Croft, 2004). Probabilistic models of cognitive linguistics are described in (Chater, 2006). An insightful introduction to human cognitive abilities is available in (Carroll, 1993).

2.2 Motivation

We strive to better model the conceptual linkage among words in a query sentence, such that the linked semantic field (query subnet) can be searched for in a larger network of RDF based

We abstract the problem to the level of cognitive semantics, by making use of the concept of semantic *forms*. According to existing cognitive psychology, *forms* are used by the human psyche to process and store semantic information structures in the brain (Carroll, 1993). *To the best of our knowledge, computationally modelling semantic forms borrowed from the domain of cognitive psychology has not been previously used in semantic query understanding in the domain of natural language processing*.

3 The Cognitive Structure-of-Intellect Model

The main hypothesis proposed by the Structure-of-Intellect model is that *human cognition is robust to noisy sentence constructions because it strives to detect semantic information at different levels of granularity*. The noisier the sentence, the coarser is the granularity of semantic information detection employed by the human cognition. In this section, we qualitatively introduce the different semantic *forms* from Guildford's structure-of-intellect cognitive model and describe how *form* interaction patterns play a key role in semantic subnet extraction.

3.1 Granular Hierarchies in Semantic Information

Semantic *forms* consist of five entities that capture the structure of information contained within a natural language sentence as perceived by the human cognition. A remarkable thing about semantic *forms* is that they are structured as granular hierarchies (i.e. one *form* is composed of other *forms*). Following is a description of the semantic *forms* starting with finer granularity:

Unit: Every item of a query sentence can be regarded as part of some chunk, of which *units* are the most basic entities. *Units* will cover most words of a sentence, from intangible ideas like 'love' to tangible objects like 'cars'. For example, the name 'Anna Chakvetadze' is a *unit*. The cognition of semantic *units* has to do with one's vocabulary (Guilford, 1977).

Class: When *units* have one or more attributes in common, they can be grouped in *classes*. *Units* belonging to a *class* will share connectivity to at least one common attribute node. *Classes* can be narrow or broad. For example, the *unit* 'Anna Chakvetadze' can belong to the very broad *class* 'female', a moderately broad *class* 'Russia' or a narrow *class* 'Tennis'. The size of the *class* (narrow/ broad) qualitatively determines the size of the search space for related concept retrieval.

Relation: *Relations* are *kinds* of connections between *units*. When any two entities are connected in the semantic network, there are three items of information involved – two *units* and the *relation* between them. *Relations* between search keywords play an integral role in realizing *class* or *unit* interconnections in the query. For example, 'Steffi Graf' and 'Andre Agassi' could be connected by the *relation*: *married*, while both belonging to the *class: tennis players*.

System: A *system* is the most complex item in semantic information. *Systems* are composed of more than two interconnected *units*. *Systems* may also comprise of overlapping *classes*, multiple interconnecting *units* and diverse *relations*. They often occur as an order or sequence of *units*. Add 'Maria Sharapova' and 'Sasha Vujacic' to the previous example of 'Steffi Graf' and 'Andre Agassi', and we get a *system: married sportspersons*.

Word	Thursday	witches	market	driving	mansion	school
Form	*unit*	*class*	*system*	*relation*	*Unit*	*system*

TABLE 1 – Examples of *forms* attached to query words.

4 The Proposed Computational Cognitive model

In our proposed approach, consider each observed symbol as the tuple: {word, POS tag, NP chunk number}. We can employ basic sequence labeling idea here, by considering the chain of *forms* that link the tuples as hidden states. Using the training data, a CRF model (McCallum, 2000) can then assign optimal state chains to samples of observed symbols, from which we learn the kinds of *form* chains (interactions) that exist. Steps for computationally modeling the cognitive notion of semantic *forms* are described in this section.

We begin with formal definitions, followed by describing some pre-processing techniques and finally, the detailed description of model features.

4.1 Formal Definitions

Consider an NL sentence Q. Our assumption is that Q is a carrier of information. Every word is a linguistic variable in Q. It is well known that information is expressible as a restriction (i.e. a constraint) on the values that a variable can take (Zadeh, 1998). By this flow of thought, consider *W* as a constrained variable in *Q*, let *R* be the constraining relation in *Q* and ζ (zeta) represent how *R* constrains W. Then, every NL sentence *Q* can be represented as: $Q \rightarrow W\ \zeta\ R$

It is possible for *W* to be a vector-valued random variable. The primary constraint *R* is a restriction on the *form* values that can be probabilistically assigned to *W*. Hence, *W* can take up values of different *forms* from the set (*unit, class,* … , *transform*) with probabilities (p_{unit}, p_{class}, …, $p_{transform}$) respectively. Thus, *W* is constrained using the probability distribution *R* as:

$$W\ \zeta\ (p_{unit} \backslash unit +\ p_{class} \backslash class + \cdots + p_{transform} \backslash transform)$$

The singular variable *W* takes values from the universe of discourse *U*, such that values of *W* are singletons in *U*. On the other hand, the semantic *form* of *W* is a variable whose values depend on the granular collections in *U*. Said alternately; *the granular precision of a word in U is expressed through its semantic form*. The type of *form* assigned to a word depends on the cluster size of elements in *U* that have common attributes or behaviour related with the concepts of that word.

The overall process is described at an abstract level in Fig. 2, where ellipses represent the *form* of a word. Consider four key words W1, W2, W3, and W4 in the query (Q) that need to be connected as some semantic subnet. Let $Form(W1)$ denote the *form* associated with the word *W1*. In step (i): we are uncertain of the semantic subnet connection among the words. In (ii), our goal is to label the words with semantic *forms* to help extract the query subnet. In (iii), we use the *form* interconnection patterns (described in Section 4.3.3.2) to retrieve the connection among the *forms* for the four words when they exist together in some Q. Finally, in (iv), we can connect the words as a query subnet by shadowing the connected *form* pattern that exists among the *forms*.

known Python NL toolkit stop word list. We used the Stanford POS tagger for POS tagging. For long queries, chunking is necessary. The chunking process is inspired by (Huang, 2009).

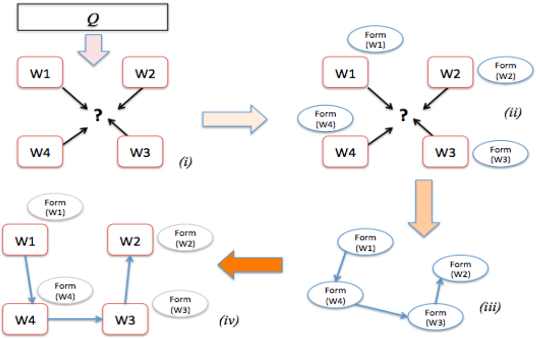


FIGURE 2 – Overview of process flow from receiving input query Q to subnet extraction.

Chunking: Consider Q to be a query sentence in natural language L containing words belonging to the vocabulary set V. Let S_Q be the sequence of POS-tagged symbols associated with Q, i.e.

$S_Q = \{s_1, s_2, \ldots, s_N\}$, where $s_n = \langle w_n, t_n \rangle$, $w_n \in V$, $t_n \in T'$ for N words in Q.

Given S_Q we can define the k^{th} chunk $(\mathcal{C}_k)$ as: $\mathcal{C}_k = (\langle w_i, t_i \rangle, \langle w_{i+1}, t_{i+1} \rangle, \ldots, \langle w_j, t_j \rangle)$ for some $i < j \leq N$ and $1 \leq k \leq M$ for a total of M chunks in the query. We assume that no two chunks have any common words, i.e. chunks are non-overlapping. Then, the task involves determining all the M chunks based on S_Q, s.t. $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_k, \ldots, \mathcal{C}_M\}$.

This generates the chunked query set: $S_{QC} = \{\langle s_1, \mathcal{C}_1 \rangle, \ldots, \langle s_l, \mathcal{C}_1 \rangle, \langle s_{l+1}, \mathcal{C}_2 \rangle, \ldots, \langle s_N, \mathcal{C}_M \rangle\}$, where $\mathcal{C}_1 = (s_1, \ldots, s_l)$, $s_l \in S_Q$, for some l, $1 \leq l \leq N$. Following similar methods as used in (Huang, 2009) and given $\{s_{i-1}, s_i, s_{i+1}\}$, we can find $p(c_1, c_2, \ldots, c_N \,|S_Q)$ as:

$$p(c_1, c_2, \ldots, c_N \,|S_Q) = \Psi \prod_{i=1}^{N} p(s_{i-1}|s_i, c_i)\, p(s_{i+1}|s_i, c_i)\, p(s_i|c_i)\, p(c_i) \qquad (1)$$

$$\Psi = {}^{1}\!/_{p(s_1, s_2)\, p(s_2, s_3) \ldots, p(s_{N-1}, s_N)}$$

and c_i represents if s_i is inside, outside or start of some NP chunk. The individual probabilities of Eq. (1) can be estimated from the training set.

4.3 *Form* Tagging Using CRFs

The task of tagging words of a sentence with semantic *forms* from the set of *forms* (F) leverages a CRF model. The result is the set S_{QF} of *form* labelled words. First, we briefly describe CRF in the light of our problem, followed by feature functions and learning weights.

4.3.1 Conditional Random Fields

A *state feature* is an element of $\boldsymbol{l_f}$ of the structure $state(y, \boldsymbol{x}, i)$ where *i* is the input position, *y* is a label and $\boldsymbol{x}$ is the input sequence. A *transition feature* is an element of l_f of the structure $tran(y, y', \boldsymbol{x}, i)$ where *y*, y' are labels.

The global feature vector for an input sequence $\boldsymbol{x}$ and a label sequence $\boldsymbol{y}$ is:

$$\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x}) = \sum_i \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{x}, i) \qquad (2)$$

Individual feature functions are described in Section 4.3.2. A conditional distribution that obeys the Markov property, which is: $p\left(Y_i \middle| \{Y_j\}_{j \neq i}, \boldsymbol{X}\right) = p(Y_i | Y_{i-1}, Y_{i+1}, \boldsymbol{X})$ can be written as:

$$p_{\boldsymbol{\lambda}}(\boldsymbol{Y}|\boldsymbol{X}) = \frac{\exp\{\boldsymbol{\lambda}.\boldsymbol{F}(\boldsymbol{Y},\boldsymbol{X})\}}{Z_{\boldsymbol{\lambda}}(\boldsymbol{X})} \qquad (3)$$

where $Z_{\boldsymbol{\lambda}}(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \exp\{\boldsymbol{\lambda} \,.\, \boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})\}$.

Note the denominator of Eq. (3) is independent of $\boldsymbol{y}$. Then the most probable sequence of *form* labels *($\boldsymbol{y}$*)* for the input sequence $\boldsymbol{x}$ is:

$$\boldsymbol{y}^* = \arg max_{\boldsymbol{y}}\, p_{\boldsymbol{\lambda}}(\boldsymbol{y}|\boldsymbol{x}) = \arg max_{\boldsymbol{y}}\, \boldsymbol{\lambda}.\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x}) \qquad (4)$$

Eq. (4) can be solved using the Viterbi Algorithm (McCallum, 2000).

4.3.2 Feature Functions

Feature functions are key components of CRF (see Fig. 3). The general structure of a feature function is $z(f_{i-1}, f_i, \boldsymbol{x}, i)$ which looks at two adjacent states f_{i-1}, f_i, the whole input sequence $\boldsymbol{x}$ where *i* is the current location in this sequence, and assigns some weight. They can be defined in different ways, e.g., we have a feature like: if the current word is 'Nile' and the current state is '*unit*' then we give the feature a positive weight, otherwise not. Each feature function has a binary output and can take as inputs the value of a feature and particular values of the current *form* f_i and the previous *form* f_{i-1}. We use the training corpus queries to build the atomic feature set for the CRF. Let F_u, F_r, F_c, F_s represent *unit, relation, class* and *system respectively*.

In the examples below, a binary value of 1 indicates the presence of the feature, and 0 the lack of the feature. '⋏' denotes logical AND. We implement four types of binary atomic features:

(1) *Simple Feature Function*: A simple feature function depends only on a word and its connected *form*. For example,

$$z(f_{i-1}, f_i, \boldsymbol{x}, i) = \begin{cases} 1, & if\ \ x_i =' \ music' \ \curlywedge f_i = F_s \\ 0, & otherwise \end{cases}$$

(2) *Overlapping Feature Function*: An overlapping feature function depends on *form* of a word and on its successor word. Under normal conditions, Hidden Markov Models (HMMs) are unable to realize overlapping features (unlike CRFs). A suitable example would be:

$$z(f_{i-1}, f_i, \boldsymbol{x}, i) = \begin{cases} 1, & if\ x_i =' \ and' \ \curlywedge f_{i-1} \neq F_r \\ 0, & otherwise \end{cases}$$

(3) *Form Transition Feature Function*: A *form* transition feature function depends on successive *forms* such as:

$$z(f_{i-1}, f_i, \boldsymbol{x}, i) = \begin{cases} 1, & if\ f_{i-1} = F_r \curlywedge f_i = \{F_u, F_c\} \end{cases}$$

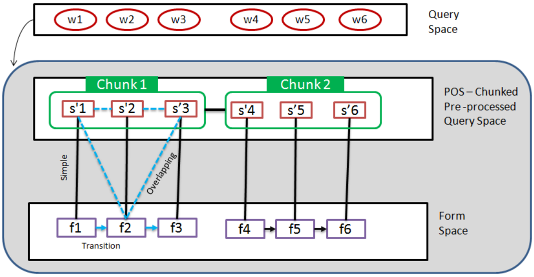


FIGURE 3 – Different features functions in the CRF model: from query words to *form* labeling.

In Fig. 3, each '*s*' element in the POS-chunked pre-processed query space represents a tuple <word, POS, NP Chunk number>. There are 828 atomic features in our system, obtained from words in the vocabulary and shifted-conjugation patterns.

This initial feature set is then grown using feature induction (McCallum, 2003), resulting in a total of 23,713 features. A quasi-Newton method is used to adjust all parameters of the CRF model to increase the conditional likelihood. When training the CRF, we use pre-conditioning to ensure fast convergence of the conjugate gradient method (Sha, 2003). On average, our technique requires 12-13 forward-backward iterations to reach an objective function value, which is in close proximity (~96%) to the maximum.

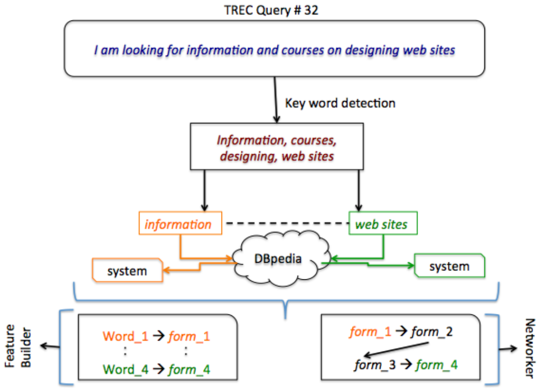


FIGURE 4 – Feature Builder and Networker learning from training queries.

4.3.3 Feature Generation

4.3.3.1 State features

The CRF labeller’s state feature set is assorted using a feature builder. The feature builder is trained using a seed set of words and their related *forms* obtained using DBpedia RDF resource (Fig. 4). DBpedia contains structured information collected from Wikipedia and has been extensively used for semantic analysis of content (Auer, 2007). The RDF semantic network of DBpedia is arranged in granular categories. Thus, for every node (which represents a concept word), we can calculate the normalized degree centrality, which gives us an estimate of the generality of the node. The more general concept nodes have higher centralities (Coursey, 2009).

Fig. 4 illustrates how words and their tagged *forms* are collected from a training query. Keywords are identified from a query by stemming and eliminating stop words. Using DBpedia to classify the word into a semantic *form* follows this. The vocabulary containing words → *forms* is updated as more training examples are seen and used by the feature builder.

4.3.3.2 Transition features

Given enough training samples of the sentence *Q*, the variable *W* and constraint *R*, we can deduce the pattern ζ, which identifies how *R* constrains *W*. This pattern ζ contains information about the ordering in *Q* with respect to *W* (recall W could be vector-valued) such that they are mapped to *R*. That is to say, every *form* has some specific interaction pattern with other *forms* when they exist together/adjacent in *Q*. This interaction pattern among *forms* in *U* is signified by ζ. Interaction patterns provide insight into the question: how are three or more *forms* connected when appearing in *Q*? For example, if we see words {*A, B, C*} having *forms* {*relation, class, unit}* respectively, then would the query subnet be of the ordering A-B-C, B-A-C or C-A-B?

The networker learns interaction patterns at the chunk level, modelling each chunk as a potential branch for a rooted semantic query subnet. This viewpoint is derived from the observation that branches of most annotated query subnets are composed of individual or contiguous chunks of the original query. The *form* interaction set $\tilde{F}$ is a simple ordered set: $\{(f_i, f_j)\}$, where $f_i, f_j \in F$ representing a complete or part of a directed chain $f_i \rightarrow f_j$. We only use *forms* connected within a chunk to populate the set $\tilde{F}$. Fig. 5 shows results collected using a Trellis diagram.

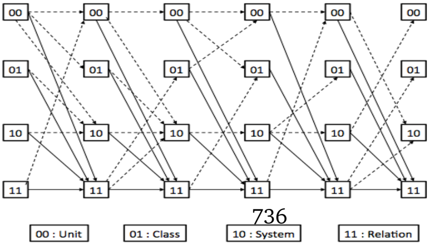


FIGURE 5 – Trellis diagram of possible Viterbi paths representing sequence of labeled *forms*.

measure: $z(f_{i-1}, f_i, \boldsymbol{x}, i) = 1\ if\ f_i = b\ and\ f_{i-1} = a$ and assign weights to this feature based on *K* training subnet samples as:

$$w_{a,b} = 1 + \log[\ \textstyle\sum_{1 \leq k \leq K} n(a,b) * |k|/K\ ] \tag{5}$$

where *k* is the *kth* sample in the training set of subnets, *n(a,b)* = 1 if the *form* transition $a \rightarrow b$ appears as some edge of the *kth* sample subnet and |*k*| is the length of the subnet branch containing $a \rightarrow b$. If the *form* transition $a \rightarrow b$ is not present, then *n(a,b)* = 0.

Eq. (5) achieves a simple goal: it takes the human labelled subnets into consideration while allocating weights for transition feature functions. The more we notice a particular *form* transition repeated, the higher the weight it is given as a potential transition feature.

From the *form* interaction patterns $\tilde{F}$ and the chunk set $\mathcal{C}$, the networker builds a set $S_{QF} = \{\langle s_1, f_1\ \rangle, \dots, \langle s_3, f_3, \rangle, \dots, \langle s_6, f_6, \rangle\}$ structured as a tree $G = (S_{QF}, E)$ where $E \in \tilde{F}$. If $|\mathcal{C}_i|$ represents the number of POS-tagged symbols in some chunk $\mathcal{C}_i \in \mathcal{C}$, then the height of subnet is $\leq \max(|\mathcal{C}_i|)$, taking into account that consecutive *units* within a chunk may be collapsed into a single node. G is the semantic subnet.

5 Experiments

5.1 Data Description, Evaluation Metrics and Benchmarks

Data: We test our model on each of these three datasets: (a) The TREC 2011 (TREC) web topic dataset has 50 topics (Clarke, 2011). Each topic has 1-7 queries associated with it. All queries within a topic resemble similar search intent. There are a total of 243 queries in the TREC topic dataset. 77% of the queries in the TREC dataset have 11-14 words. (b) The Microsoft Question Answering Corpus (MSQA), which is aimed at querying documents belonging to the Encarta-98 encyclopedia (MSQA, 2008). There are 1365 usable queries in this dataset and 85% of the queries have 5-10 words. (c) The last dataset consists of ~ 3400 raw search query feeds collected from a commercial web search engine (denoted as 'WSE'). Queries containing 4-20 words are chosen for evaluation. The distribution of average number of words per query is shown in Fig. 6.

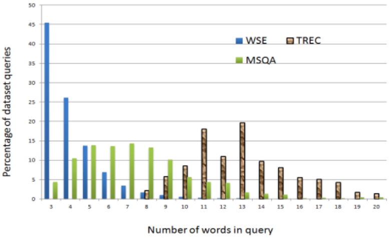


FIGURE 6 – Distribution of avg. number of words per query in the three datasets.

The three datasets represents gradually rising levels of challenge in terms of query construction diversity, number of words in query and interpretability, with TREC being the least diverse and

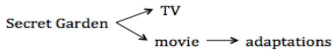
Query ID	Query Sentence	Query Subnet
TREC 43	*Find reviews of the various TV and movie adaptations of The Secret Garden*	

TABLE 2 – Example of query subnet generated from query.

Several other small optimizations are implemented: (a) we collapse consecutive *units* into a single *unit* when creating the subnet. (b) We use a simple root selection algorithm: when only *relation* words are found connecting two chunks $\mathcal{C}_i , \mathcal{C}_j \in \mathcal{C}$, we search $\mathcal{C}_{j+1}$ for *units* or *classes*. If $\mathcal{C}_{j+1}$ lacks a *unit* or *class*, we search $\mathcal{C}_{i-1}$instead. For example, in the query #TREC43 (Table 2), '*various TV'* and '*movie adaptations'* are connected by the conjunction '*and'*. Therefore, we search in $\mathcal{C}_{j+1} =' of\ The\ Secret\ Garden'$ and since we find a sequence of two *units* ('Secret', 'Garden'), we collapse it to a single *unit* and represent it as root.

We used annotators to hand label the queries in the datasets to build query subnet trees. The inter-annotator agreement on subnet structure was 72.3%. Disagreements were limited to just 1 node position in 82% disagreed cases. Thus, we consider this hand labelled set as the gold standard for comparing the machine generated subnet.

Metrics: Since our output (query subnet) is a tree where each node belongs to the set of query words, a 'tree-likeness' metric is essential to judge quality of results produced in terms of *structure*. We use *Consistency Index* (*CI*) as a metric to judge the quality of the subnet generated (Cardona, 2009). Mathematically, *CI* can be defined as:

$$CI = node\ position\ matches\ between\ T1\ \ and\ T2/\#\ nodes\ in\ T2$$

where, *T1* represents the query subnet tree generated by a machine algorithm, *T2* is the query subnet tree of the gold standard and # represents the number of nodes. In (Booth, 2009), the authors evaluate their subnets using simple measures like 'nodes correctly resolved' or 'semi-correctly resolved'. However, we believe that *CI* captures the effect of *structural relatedness* more intuitively. Table 3 lists the average *CI* values obtained for various datasets for the proposed approach and the comparison benchmarks.

Benchmarks: We compare the proposed model (*formNet*) against 3 benchmarks. We test our model against (a) the POS based approach introduced in (Booth, 2009) for generating subnets from query sentences (called *posNet*), (b) a non-*form* CRF (denoted as *nfCRF*) used in (Sha, 2003), whose features are based on POS only, and (c) a non-chunked version of our model (denoted as *noChnk*), to compare the gain due to semantic *forms* vs. chunking.

5.2 Test Results

We measure the average CI for queries in each dataset with our technique against the above benchmark techniques. Results are reported in Table 3. For each dataset, we provide a detailed bar graph describing percentage of queries that produced outputs in some particular CI range.

TREC: Fig. 7(A) shows that *formNet* achieves *CI*=1 for 63.1% queries. In fact, only 9.2% of the

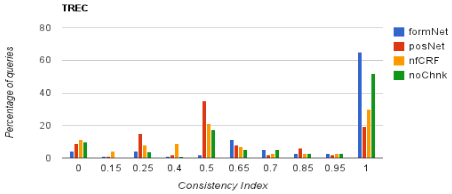


FIGURE 7. (A) – Percentage of queries that yielded some CI for TREC

	posNet	*formNet*	nfCRF	noChnk
TREC	54.8	***83.6***	58.6	74.0
MSQA	53.6	***79.5***	43.3	77.3
WSE	48.2	***73.2***	36.1	68.4

TABLE 3 – Evaluation results for various datasets in terms of average CI (%)

MSQA: Table 3 shows that *formNet* provides average *CI*=0.795 for MSQA queries whereas the benchmark *posNet* produces an average *CI*=0.536. This signifies ~ 49% improvement in performance. Fig. 7(B) shows that *formNet* can retrieve 55% queries with perfect match and produces a *CI*>0.5 for 85% queries in the dataset. In contrast, the benchmark *posNet* could only produce *CI*>0.5 for 38% queries. TREC queries are grammatically richer than MSQA; therefore a drop in overall performance is expected when evaluating MSQA. Interestingly, *forms* seem to be playing a stronger role in MSQA, since a traditional CRF performs poorly in this case.

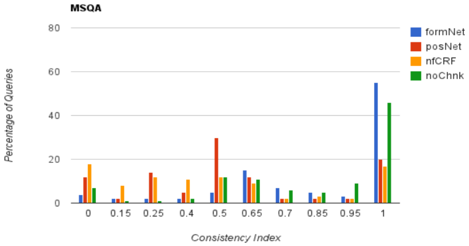


FIGURE 7. (B) – Percentage of queries that yielded some CI for MSQA

WSE: WSE queries are most diverse in construction and number of words. In Fig. 7(C), we see that performance is reduced for all techniques, but *formNet* still performs better than *posNet* by 51.86%. Observe that *noChnk* performs worst for TREC when compared to *formNet* than for any other dataset as indicated in Table 3 (difference between average CI for *formNet* and *noChnk*). This reaffirms our previous observation from the query data: TREC queries consist of longer substantially better than *noChnk* for MSQA and WSE datasets, whereas no chunking for TREC significantly deteriorates performance. This indicates that chunking has a stronger impact in TREC, a dataset where 77% queries have more than 11 words (Fig. 6). In comparison, only ~10.8 % queries in MSQA and 7.6% queries in WSE have more than 9 words.

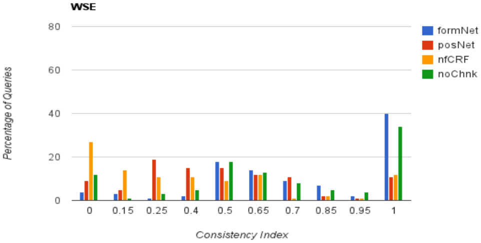


FIGURE 7. (C) – Percentage of queries that yielded some CI for WSE.

Cross Dataset Testing: Different datasets differ in query structure, context and length of query. To ensure robustness to different training environments, we perform cross dataset testing, i.e., train on one dataset and test on another (read TRAIN_TEST). Here, we report *formNet* performance. The average *CI* achieved by *formNet* is as follows: TREC_MSQA: 0.53, TREC_WSE: 0.44, MSQA_TREC: 0.68, MSQA_WSE: 0.58. We can observe that cross dataset testing provides best results when we train on MSQA and test on TREC. This is potentially due to the fact that the TREC dataset query structures are quite limited in construction, which are contained within queries of MSQA. Performance is worst when we train on TREC and test on WSE. This is potentially due to the diverse and noisy queries in WSE not captured during limited training over TREC. Nevertheless, for MSQA_WSE, *formNet* retrieves query subnets with *CI* > 0.5 in 73.1% cases and *CI* > 0.75 in 33% cases, suggesting robustness of *formNet* to web scale.

6 Conclusion

Several papers on computational cognitive psychology dwell on the fact that cognitive psychology models cannot be purely verified on the basis of behavioural experiments (Chater, 2006). For researchers in the domain of NLP, a fascinating possibility is to model cognitive techniques computationally and test their robustness to noise in NL. Natural languages are undeniably imprecise, especially in the realm of semantics. The primary reason of this imprecision is the fuzziness of class boundaries (Zadeh, 1998). Surprisingly, robustness to imprecision is often achieved by slightly relaxing the rigidity imposed by lexical grammar, by means of parsing at a higher abstraction than POS.

In this paper, we reproduce the structure-of-intellect model of cognitive psychology computationally. Exploring the various interactions among the semantic *forms* provides insights

References

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. (2011). Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Research* (November 2011), 2493-2537.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning* (ICML '01), San Francisco, CA, USA, 282-289.

Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. (2000). Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning* (ICML '00), San Francisco, CA, USA, 591-598.

Minhua Huang and Robert M. Haralick. (2009). Identifying Patterns in Texts. In *Proceedings of the 2009 IEEE International Conference on Semantic Computing* (ICSC '09). IEEE Computer Society, Washington, DC, USA, 59-64.

Joel Booth, Barbara Di Eugenio, Isabel F. Cruz, and Ouri Wolfson. (2009). Query Sentences as Semantic (Sub) Networks. In *Proceedings of the 2009 IEEE International Conference on Semantic Computing* (ICSC '09). IEEE Computer Society, Washington, DC, USA, 89-94.

Rebecca Bruce and Janyce Wiebe. (1994). Word-sense disambiguation using decomposable models. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* (ACL '94). Association for Computational Linguistics, Stroudsburg, PA, USA, 139-146.

Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. (2003). Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th international conference on Intelligent user interfaces* (IUI '03). ACM, New York, NY, USA, 149-157.

E. Kaufmann, A. Bernstein and L. Fisher. (2007). “NLP-Reduce: A “naïve” but domain-independent natural language interface for querying ontologies. In *4th European Semantic Web Conference (ESWC).*

Gabriel Cardona, Francesc Rossello, and Gabriel Valiente. (2009). Comparison of Tree-Child Phylogenetic Networks. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* 6, 4 (October 2009), 552-569.

Fei Sha and Fernando Pereira. (2003). Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1* (NAACL '03), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA, 134-141.

Lance A. Ramshaw and Mitchell P. Marcus. (1995). Text chunking using transformation-based

Language Processing. *MIT Press*.

Joy P. Guilford. (1977). Way beyond the IQ. *Creative Education Foundation, NY.*

Soren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann and Zachary Ives. (2007). DBpedia: A Nucleus for a Web of Open Data. In *6th Inter. Semantic Web Conference,* 11-15.

J Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. (2009). Understanding user's query intent with wikipedia. In *Proceedings of the 18th international conference on World wide web* (WWW '09). ACM, New York, NY, USA, 471-480.

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. (2009). Named entity recognition in query. In*Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '09). ACM, New York, NY, USA, 267-274.

Roberto Navigli. (2009). Word sense disambiguation: A survey. *ACM Comput. Surv.* 41, 2, Article 10 (February 2009), 69 pages.

John B. Carroll. (1993). Human Cognitive Abilities. *Cambridge University Press, Cambridge.*

Andrew McCallum. (2002). Efficiently inducing features of conditional random fields. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence* (UAI'03), Uffe Kjærulff and Christopher Meek (Eds.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 403-410.

William Croft and D. Alan Cruse. (2004). Cognitive Linguistics. *Cambridge University Press.*

N. Chater, J. B. Tenenbaum and A. Yuille. (2006). Probabilistic models of cognition: Conceptual foundations. In *Trends in Cognitive Science,* 10.

A. Herdagdelen, M. Ciaramita, D. Mahler, M. Holmqvist, K. Hall and S. Riezler. (2010). Generalized Syntactic and Semantic Models of Query Reformulation. In *SIGIR*, 283-290.

Lotfi A. Zadeh. (1998). Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. In Soft Computing 2(1): 23-25.

MSQA. (2008). http://research.microsoft.com/en-us/downloads/88c0021c-328a-4148-a158-a42d7331c6cf/

Charles A. Clarke, Nick Craswell, Ian Soboroff and Ellen M. Voorhees. (2011). Overview of the TREC 2011 Web Track. *Text Retrieval Conference*.

Silviu Cucerzan. (2007). Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning,* pp 708-716, June 2007.

David McClosky, Mihai Surdeanu, and Christopher D. Manning. (2011). Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for*

Association for Computational Linguistics, Stroudsburg, PA, USA, 326-334.

Kino Coursey and Rada Mihalcea. (2009). Topic identification using Wikipedia graph centrality. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers* (NAACL-Short '09). Association for Computational Linguistics, Stroudsburg, PA, USA, 117-120.