

Scalable Multi-Dimensional User Intent Identification using Tree Structured Distributions

Vinay Jethava
Chalmers University
Gothenburg, Sweden
jethava@chalmers.se

Liliana Calderón-
Benavides
UNAB
Bucaramanga, Colombia
mcalderon@unab.edu.co

Ricardo Baeza-Yates
Yahoo! Research
Barcelona, Spain
rbaeza@acm.org

Chiranjib Bhattacharyya
Indian Institute of Science
Bangalore, India
chiru@csa.iisc.ernet.in

Devdatt Dubhashi
Chalmers University
Gothenburg, Sweden
dubhashi@chalmers.se

ABSTRACT

The problem of identifying user intent has received considerable attention in recent years, particularly in the context of improving the search experience via query contextualization. *Intent* can be characterized by multiple dimensions, which are often not observed from query words alone. Accurate identification of *Intent* from query words remains a challenging problem primarily because it is extremely difficult to discover these dimensions. The problem is often significantly compounded due to lack of representative training sample. We present a generic, extensible framework for learning the multi-dimensional representation of user intent from the query words. The approach models the latent relationships between facets using tree structured distribution which leads to an efficient and convergent algorithm, FastQ, for identifying the multi-faceted intent of users based on just the query words. We also incorporated WordNet to extend the system capabilities to queries which contain words that do not appear in the training data. Empirical results show that FastQ yields accurate identification of intent when compared to a gold standard.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: *Information Search and Retrieval - Search process*; G.3 [Probability and Statistics]: *Probabilistic algorithms*

General Terms

Algorithms, Performance, Theory

Keywords

web search, query intent, facets, Chow-Liu, WordNet, FastQ

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR '11, July 24–28, 2011, Beijing, China.

Copyright 2011 ACM 978-1-4503-0757-4/11/07 ...\$10.00.

1. INTRODUCTION

Identifying user intent¹ behind search queries plays a crucial role in providing a better search experience [16, 29, 28]. The traditional classification of user intent as *navigational*, *transactional* or *informational* has focused on intent as a single dimension [3] or hierarchical extensions [26, 2, 14].

There is a need for a systematic framework to analyze user intent in terms of multiple dimensions (or *facets*) of interest. In fact, if a facet can be predicted well, it is possible to contextualize the answer for a query, triggering a different ranking algorithm or even a completely different user experience. For example, an informational query will rank better information coming from trusted web sites or a time sensitive query may give better ranking to news. Similarly, a local query may trigger a map or a genre specific query may show a particular screen layout. The multiple facets of interest might be correlated, which poses a challenge compared to past approaches [19, 2, 15]. Further, one should be able to predict the user intent for new queries in a reasonably fast manner which can be incorporated into existing search engine technology [8].

This paper tries to model the probabilistic dependencies between the various *dimensions* underlying the user intent of a search query, and then proposes to use the model to identify the *intent* behind the query. Our hope is that by modeling the dependencies well, it might yield a more accurate identification of the underlying intent. We use the set of facets from [4] to illustrate our method. As our model is more complex, we explore the potential of our technique using the minimal set of features available, that is, just the query words. Using more features, such as clicks, would just improve our results.

Our approach immediately raises several challenging questions, namely (a) What kind of models should one use in modeling dependencies? (b) How does one model the joint distribution of query words and the dimensions? and (c) Is it possible to augment the model to cater for unseen query words?

This paper poses the problem of identifying the user's intent behind a query as an *inference problem over a tree structured graphical model*. We argue that the tree structured

¹In this paper we use intent in a broad sense, including generic and specific intents.

model serves as a natural choice for discovering latent dependencies between several *dimensions* of a query as it is the *simplest* distribution which goes beyond the independence assumption. The associated learning problem is solved by using the Chow-Liu algorithm [6, 21, 27]. In addition, the resultant algorithm should be scalable and preferably online for real world use.

Most of the works available in the literature are based on the analysis of query logs and click-through data. However, the additional information might not be available, or restricted due to privacy concerns. Our method infers the user intent from the query words only, without using click-through documents, web session information, and other meta-data. This shows the potential of the algorithm, as with more features, the performance should improve.

We model the query as a factorial distribution over the observed *query words*, which is equivalent to making the assumption that the impact of a word in the query is independent of the other words. This results in a simplified model which allows an online algorithm for user's intent identification. We also incorporate WordNet [12], a lexical database, to improve the classification accuracy of the user intent of search queries for unseen query words.

In order to perform the identification of the underlying user's needs in the described setting, we introduce *FastQ*, an online distributed algorithm for classifying user intent for search queries. The algorithm dynamically constructs a tree structured factor graph [18] with the appropriate distribution. The intent is inferred by using standard belief propagation (BP) algorithms, and the tree structure of the dynamically constructed factor graph guarantees the convergence of the BP algorithm. Further, the small size of the resulting factor graph corresponding to a search query allows fast (approximate) inference and allows online classification of the user's intent. Experimental timing results show that the method can be incorporated into existing search engine technology [8, 5].

Our Contribution. The main contributions of this work are:

1. We cast the problem of identifying user intent for web search queries as an inference problem; and present a tree structured graphical model for modeling the user intent based on just the query words,
2. The resulting inference is difficult; and we present a scalable, online algorithm *FastQ* for classifying user intent for search queries, and
3. We incorporate WordNet [12], a lexical database, to identify new words present in the search query which are absent in the existing training vocabulary.
4. Our framework is generic and naturally lends itself to extension using other facets or query features, thus enabling its use in other settings.

The rest of this paper is organized as follows: Section 2 outlines previous work related to ours. Section 3 describes our query intent model for the web query classification; and the algorithm for the online query classification; and Section 4 presents our experimental setup and the results obtained. Finally, Section 5 presents our conclusions.

2. RELATED WORK

User's intent. According to Broder's taxonomy [3], the user intents can be *informational*, *navigational* and *transactional*. Albeit this taxonomy was further extended by Rose & Levinson [26], the original categories are the most widely used, and have been the basis for an important number of studies in the IR area. In the same context, Baeza-Yates *et al.* [2] have proposed that the user intents can be classified as *informational*, *not-informational* and *ambiguous*. Hu *et al.* [14] have interpreted the user's query intent through the concepts included in Wikipedia; the authors exemplify their proposal throughout the identification of the following intents: travel, personal name, and job finding. More recently, Calderón-Benavides *et al.* [4] have studied the user's intent as a multi-dimensional composition of facets.

Sources to identify the user's intent. In their work, Kang & Kim [17] used information from document content, links and URLs to classify the user's queries into topic relevance task (informational intent), homepage finding task (navigational intent), and service finding task (transactional intent). Lee *et al.* [19] took into account the past user-click behaviour and the anchor link distribution to determine the navigational and informational intents of queries. The content of the documents selected by the users in answer to a query is used in [2]. Liu *et al.* [20] exploited click-through data in order to find navigational and informational/transactional type queries.

Nguyen and Kan [23] defined functional *facets* to describe the user's intent. In their work, Ashkan *et al.* [1] developed a methodology to determine commercial/non-commercial and navigational/informational intent of queries by using ad click-through features, query features, and the content of search engine result pages. In [14], the authors have mapped the user's query intent into the concepts included in Wikipedia, and have represented them through the articles and categories related to each concept.

Recent work on query personalization has focused on modeling the query intent [9, 29, 24, 28] and user behaviour [10, 31]. In particular, Teevan *et al.* [29] investigated the variability in user intent and the relevance search results obtained in terms of implicit relevance measures based on click-through data. They use interdependent sets of features defined using information available in (i) the queries (e.g. *query strings*, *special syntax*, etc.); (ii) history of similar queries (e.g. $\#\{\text{queries issued}\}$, etc.); and (iii) query results (e.g. *click entropy*, etc.). Our work can be viewed as a probabilistic framework for semi-supervised inference of multi-dimensional user intent using *query strings* and the history information available in the queries.

Methods to determine the user's intent. Although early research was based on the manual classification of small sets of queries [3, 26], the use of some of the existing machinery from IR, machine learning, as well as natural language processing techniques have allowed an important progress towards determine the user query intents in bigger datasets. In this context, [17] calculated the probability that the task of a user query is topic relevance (informational queries), or homepage finding (navigational queries), based on the occurrence patterns of the query terms in web pages. Lee *et al.* [19] made an automatic identification of goals (navigational and informational) through the application of heuristics over clicks made by the users on the results offered by the search engine. In [2] supervised and unsupervised learning meth-

ods were applied in order to determine the user goals in context of ODP² categories for a set of six thousand queries from a vertical search engine. Jansen *et al.* [15] presented a rule-based automated classification system for queries into hierarchical level of informational, transactional, and navigational intent. However, a clear principle for deriving new rules is not presented and the extension of the rules for a new system of classification or another language is not clear. In [14] they use the Wikipedia link graph to represent the relations among the concept articles and the relations among articles and categories. Such a graph is traversed randomly to obtain a vector with the probabilities that each concept belong to a intent.

Existing methods use multiple features (click-through, weblogs, etc.) to identify the intent either as a simple hierarchy within the framework of *navigational*, *transactional* and *informational* goals. To the best of our knowledge, ours is the first work that identifies user’s intent in terms of multiple meaningful dimensions using automated server-side classification. The multi-dimensional based description encapsulates additional information about the user intent in a number of related dimensions, instead of a hierarchical scheme. As a difference to other proposals, in this work the identification of the user’s intent does not include click-through data or the document content, but is done by using solely the query words. Additionally, the task of identifying the user intent is performed in an online setting, creating a technology that can be incorporated into existing search engines environments with reasonably fast response time.

3. PROBLEM DEFINITION

3.1 Dimensions of the User’s Intent

We chose the multi-dimensional description of user intent presented in [4]. A description of the dimensions, as well as the values that each one of them can take, is presented in Table 1. The multi-dimensional based description encapsulates additional information about the user intent in a number of related dimensions, instead of a hierarchical scheme.

3.2 The Query Classification Problem

Let $\check{q} = [q_1, \dots, q_m]$ denote a query of length $|\check{q}| = m$ consisting of distinct³ query words $\{q_1, \dots, q_m\}$. We assume that the user intent can be encapsulated in terms of K facets e.g. *Genre*, *Scope*, etc., where \mathcal{F}_k denote the state space of the k^{th} facet e.g. $\{Yes, No\}$ for the facet *Time Sensitivity*.

Given the training data $\mathcal{D} = \{(\check{q}^{(n)}, \vec{f}^{(n)})\}_{1 \leq n \leq N}$ consisting on N queries $\check{q}^{(n)}$ and the corresponding user intent (in terms of facets) $\vec{f}^{(n)}$, we cast the problem of finding the user intent $\hat{f}^\dagger \in \mathcal{F}$ for a new query \check{q}^\dagger as an inference problem

$$\hat{f}^\dagger = \arg \max_{\vec{f} \in \mathcal{F}} P(\vec{F} = \vec{f} | \check{Q} = \check{q}^\dagger, \mathcal{D}) \quad (1)$$

$$= \arg \max_{\vec{f} \in \mathcal{F}} P(\check{Q} = \check{q}^\dagger | \vec{F} = \vec{f}, \Lambda) P(\vec{F} = \vec{f} | \Theta) \quad (2)$$

where $\mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2 \times \dots \times \mathcal{F}_K$ denotes the state space of facet classifications corresponding to a query; $\Lambda = \Lambda(\mathcal{D})$ and $\Theta = \Theta(\mathcal{D})$ are parameters based on the training data.

²<http://dmoz.org>

³Duplicate instances of a word in a query are removed during pre-processing.

Dimension	Values & Meaning
Genre	{ <i>News, Business, Reference, Community</i> }. It is considered a meta-category that provides a generic context to the user query intent.
Topic	{ <i>Adult & Sex, Arts & Culture, Beauty & Style, Cars & Transportation, Computers & Internet, Education, Entertainment, Music & Games, Finance, Food & Drink, Health, Home & Garden, Industrial Goods & Services, Politics & Government, Religion & belief systems, Science & Mathematics, Social Science, Sports, Technology & Electronic, Travel, Work</i> }. This list of topics was created by taking the first level categories included in ODP, Yahoo!, and Wikipedia.
Task	{ <i>Informational, Not Informational, Both</i> } [2]. Considered as the primary need reflected by a query.
Objective	{ <i>Resource, Action</i> }. Represents the aim of a query, without considering the format of the information to retrieve.
Specificity	{ <i>Specific, Medium, Broad</i> }. This facet describes how specialized is a query.
Scope	{ <i>Multiple, Unique</i> }. Shows whether the query contains polysemic words or not.
Authority Sensitivity	{ <i>Yes, No</i> }. Is the query designed to retrieve authoritative and trusted answers? [23]
Spatial Sensitivity	{ <i>Yes, No</i> }. Reflects the interest of the user to obtain a resource that is related to a particular spatial location (explicit or not).
Time Sensitivity	{ <i>Yes, No</i> }. Whether the information to retrieve, involves a date or period of time.

Table 1: The dimensions or *facets* which characterize user intent as in [4].

The following section discusses the choice of Λ and Θ based on the training data; and then we discuss *FastQ*, a scalable, convergent algorithm for solving the overall inference problem.

3.3 The Query Intent Model

Now we present our generative model for search query \check{q} and corresponding user intent \vec{f} assuming queries and corresponding user intents are drawn from an unknown joint distribution $P(\check{Q} = \check{q}, \vec{F} = \vec{f})$.

A full realization of the joint distribution $P(\check{q}, \vec{f})$ is computationally prohibitive. For example, if the query words are drawn from the vocabulary \mathcal{V} with maximum query length M , there are $\sum_{m=1}^M C(|\mathcal{V}|, m)$ possible queries where $C(|\mathcal{V}|, m) = \frac{(|\mathcal{V}|)!}{(|\mathcal{V}|-m)!m!}$; and the facets have $\prod_{i=1}^K |\mathcal{F}_i|$ possible states; the full probabilistic modeling would require $O(2^{|\mathcal{V}|} \prod_{i=1}^K |\mathcal{F}_i|)$ states. Hence, we use a parametric representation to reduce the dimensionality of the problem.

We assume the dataset \mathcal{D} consists of i.i.d. samples drawn from a joint distribution

$$P(\check{q}^{(n)}, \vec{f}^{(n)} | \Theta, \Lambda) = P(\check{Q} = \check{q}^{(n)} | \vec{F} = \vec{f}^{(n)}, \Lambda) P(\vec{F} = \vec{f}^{(n)} | \Theta) \quad (3)$$

where Θ and Λ are parameters which model the dependencies between the inter-related facets (Θ), and the impact of words of the query on its facet classification (Λ) respectively.

3.3.1 Relationship between Facets

We model the dependencies between the facets using the class of second order product distributions defined on the

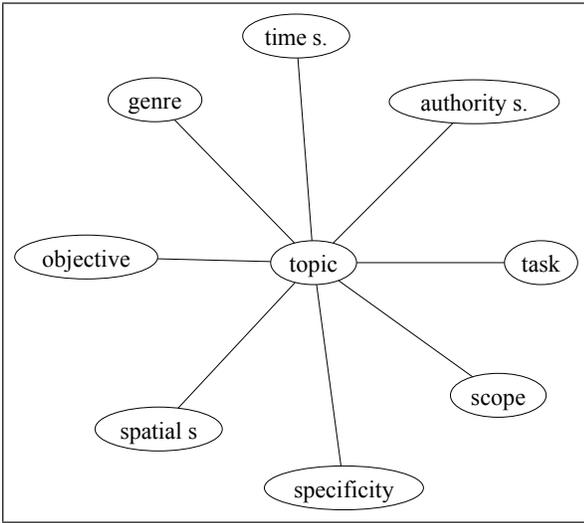


Figure 1: Example spanning tree (T) on facets.

facet variables. Let \mathcal{T} be the set of possible spanning trees defined on the facets. Then, the parameter $\Theta = \Theta_T$ defines a marginal polytope [30] on the spanning tree $T \in \mathcal{T}$. In other words $\Theta_T = \{\Theta_{ij} : (i, j) \in T\} \cup \{\Theta_i : i \in \mathcal{K}\}$ is the set of probability distributions Θ_{ij} defined on the edges of the spanning tree $(i, j) \in T$; and the probability distributions Θ_i defined on the facet variables $i \in \mathcal{K}$ such that

$$\begin{aligned} \Theta_{ij}(f_i, f_j) &\geq 0; \quad \sum_{f_i} \sum_{f_j} \Theta_{ij}(f_i, f_j) = 1 \quad \forall (i, j) \in T(4) \\ \Theta_i(f_i) &\geq 0; \quad \sum_{f_i} \Theta_i(f_i) = 1 \quad \forall i \in \mathcal{K} \quad (5) \end{aligned}$$

which satisfy the marginal consistency

$$\sum_{f_i \in \mathcal{F}_i} \Theta_{ij}(f_i, f_j) = \Theta_j(f_j) \quad \forall j \in \mathcal{F}_j, (i, j) \in T \quad (6)$$

Then, the probability distribution $P(\vec{F}|\Theta_T)$ is given by

$$P(\vec{F} = \vec{f}|\Theta_T) = \frac{\prod_{(i,j) \in T} \Theta_{ij}(f_i, f_j)}{\prod_{i \in \mathcal{K}} \Theta_i(f_i)^{d_i-1}} \quad (7)$$

where d_i denotes the degree of the i^{th} facet in the spanning tree T . Figure 1 shows a possible tree structure T defined on the facet variables.

3.3.2 Impact of Query Words on Facets

Now we describe the probability distribution of queries $P(\hat{Q} = \hat{q}|\vec{F} = \vec{f})$ conditioned on the facet variables where $\hat{q} = \{q_1, \dots, q_m\}$ denotes a query of length $|\hat{q}| = m$ and $\vec{f} = \{f_1, \dots, f_K\}$ denotes the facet states. The model assumes the following,

1. Each query has at most M distinct words. This is a valid assumption since, according to [22], nearly 70% of informational queries consists of at most 5 words, and the higher number of terms in not informational queries is 3. In our experimental data $M = 8$.
2. The length of the query $|\hat{q}|$ is distributed as $P(|\hat{q}| = m) = \beta(m)$ chosen according to [32], such that $\beta(m) \geq 0$ and $\sum_{m=1}^M \beta(m) = 1$.

3. The query words are chosen independently conditioned on the facet state \vec{f}

$$P(\hat{Q} = \hat{q}|\vec{F} = \vec{f}) = \prod_{w \in \mathcal{V}} P(w \in \hat{q}|\vec{F} = \vec{f})^{1_{w \in \hat{q}}} \quad (8)$$

where $\mathcal{V} = \{w : \exists \hat{q} \in \mathcal{D} \ni w \in \hat{q}\}$ is the vocabulary of words present in the training data.

4. The probability of a word being present in a query $P(w \in \hat{q}|\vec{F} = \vec{f})$ conditioned on the joint facet state, \vec{f} ; can be factorized into a product of probability distributions $P(w \in \hat{q}|F_k = f_k)$ of a word w being a query, conditioned on individual facet states f_k

$$P(w \in \hat{q}|\vec{F} = \vec{f}) = \prod_{k=1}^K \prod_{f \in \mathcal{F}_k} (\lambda_{w,f}^k)^{1_{\vec{f}_k=f}} \quad (9)$$

where $\{\lambda_{w,f}^k \geq 0 : w \in \mathcal{V}, k \in \mathcal{K}, f \in \mathcal{F}_k\}$ is a parameter subject to the constraints

$$\sum_{w \in \mathcal{V}} \lambda_{w,f}^k = 1 \quad \forall k \in \mathcal{K}, f \in \mathcal{F}_k \quad (10)$$

The resulting probability distribution $P(\vec{Q} = \hat{q}|\vec{F} = \vec{f}, \Lambda)$ parametrized by Λ is given as

$$P(\hat{q}|\vec{f}, \Lambda) = \frac{\beta(|\hat{q}|) e^{\{\sum_{k \in \mathcal{K}, w \in \mathcal{V}} \sum_{f \in \mathcal{F}_k} \log \lambda_{w,f}^k \cdot 1_{\hat{q}}(w) \cdot 1_{\vec{f}_k=f}\}}}{Z(\vec{f})} \quad (11)$$

where $Z(\vec{f})$ is the normalization function given as

$$Z(\vec{f}) = \sum_{m=1}^M \beta(m) \sum_{\hat{q}: |\hat{q}|=m} \exp\left(\sum_k \sum_f \sum_w \log \lambda_{w,f}^k\right) \quad (12)$$

3.3.3 Generative Model

The generative model for a search query and the corresponding user intent is presented in Algorithm 1.

Algorithm 1 The generative model for query (\hat{q}, \vec{f})

Require: \mathcal{V} {Vocabulary}
Require: Θ_T {Marginal polytope on the facets}
Require: Λ {Conditional word distribution}
 Choose $\vec{f} \sim P(\vec{f}|\Theta_T)$ as in (7) {Facet for query}
 Choose m with $P \sim \beta(\cdot)$ {Query length}
for $i = 1$ to m **do**
 Choose $q_i \sim P(\cdot|\vec{f})$ as in (11) {Generate word}
 $\hat{q} \leftarrow [\hat{q} \ q_i]$ {Append word to query}
end for
return (\hat{q}, \vec{f}) {Query and facet classification}

3.4 Estimating the Model Parameters

This section presents the procedure for computing the maximum likelihood estimation (MLE) for the model parameters $\{\Lambda, \Theta_T, T\}$ based on the dataset. The maximum likelihood estimates are obtained as a result of the optimization in

$$\arg \max_{\Lambda, \Theta, T} \sum_{(\hat{q}^{(n)}, \vec{f}^{(n)}) \in \mathcal{D}} \log P(\hat{q}^{(n)}, \vec{f}^{(n)}|\Theta, \Lambda, T) \quad (13)$$

subject to the constraints in (4), (5), (6), (10); where the probability $P(\hat{q}^{(n)}, \vec{f}^{(n)}|\Theta, \Lambda, T)$ is obtained using (3), (7)

and (11). The resulting optimization can be decoupled into separate optimization problems for (Θ^*, T^*) as

$$\begin{aligned} & \arg \max_{\Theta, T \in \mathcal{T}} \sum_{\vec{f}^{(n)}} \left\{ \sum_{(a,b) \in T} \log \Theta_{ab}(f_a^{(n)}, f_b^{(n)}) \right. \\ & \quad \left. - \sum_{a \in \mathcal{K}} \log \Theta_a(f_a^{(n)})^{d_a-1} \right\} \\ \text{s.t.} & \sum_{f_a, f_b} \Theta_{ab}(f_a, f_b) = 1; \quad \Theta_{ab}(f_a, f_b) \geq 0 \quad \forall (a, b) \in T \\ & \sum_{f_a} \Theta_a(f_a) = 1; \quad \Theta_a(f_a) \geq 0 \quad \forall a \in \mathcal{K} \\ & \sum_{f_i \in \mathcal{F}_i} \Theta_{ij}(f_i, f_j) = \Theta_j(f_j) \quad \forall j \in \mathcal{F}_j, (i, j) \in T \end{aligned} \quad (14)$$

and an optimization problem for Λ^* as

$$\begin{aligned} & \arg \max_{\Lambda} \sum_{(\check{q}^{(n)}, \vec{f}^{(n)}) \in \mathcal{D}} \left\{ -\log Z(\vec{f}^{(n)}) \right. \\ & \quad \left. + \sum_{k \in \mathcal{K}, f \in \mathcal{F}_k, w \in \mathcal{V}} \log \lambda_{w;f}^k \mathbf{1}_{[w \in \check{q}^{(n)}]} \mathbf{1}_{[f_k^{(n)}=f]} \right\} \\ \text{s.t.} & \lambda_{w;f}^i \geq 0; \quad \forall w \in \mathcal{V}, k \in \mathcal{K}, f \in \mathcal{F}_k \\ & \sum_{w \in \mathcal{V}} \lambda_{w;f}^k = 1 \quad \forall k \in \mathcal{K}, f \in \mathcal{F}_k \end{aligned} \quad (15)$$

where $Z(\vec{f}^{(n)})$ is given as (12). Now we solve the optimization problems in (14) and (15).

Notation. We introduce the function $\#(C)$ where C is a valid Boolean expression as the number of queries in dataset \mathcal{D} which satisfy the condition C , i.e. $\#(C) = \sum_{(\check{q}, \vec{f}) \in \mathcal{D}} \mathbf{1}(C)$ where $\mathbf{1}(C)$ is the indicator function, i.e., $\mathbf{1}(C) = 1$ if C is *TRUE* and 0 otherwise. For example, $\#(w \in \check{q})$ denotes the number of queries in the dataset \mathcal{D} having query word w ; $\#(f_k = f)$ denotes the number of queries for which the k^{th} facet has state f ; and so on.

3.4.1 Estimation of Facet Tree Parameters (Θ^*, T^*)

The optimization in (14) depends on the facet classifications $\vec{f}^{(n)}$ of the queries in the dataset \mathcal{D} , and not on the words $[q_1^{(n)}, \dots, q_m^{(n)}]$ in the query. Thus, the problem is to fit a second order product distribution to the multivariate data (facets) $\vec{f}^{(n)} \in \mathcal{D}$.

Chow and Liu [6] first investigated the problem of fitting a second order product distribution to multivariate data. They showed that the problem is equivalent to minimizing the Kullback-Liebler divergence [7] between the target distribution on \mathcal{D} and tree based distribution. Their algorithm shows that for the dataset \mathcal{D} and a spanning tree T ; the optimal marginal polytope Θ_T^* is given by

$$\Theta_{ij}^*(a, b) = \frac{\#(\vec{f}_i = a, \vec{f}_j = b)}{N} \quad \forall a \in \mathcal{F}_i, b \in \mathcal{F}_j \quad (16)$$

$$\Theta_i^*(a) = \frac{\#(\vec{f}_i = a)}{N} \quad \forall a \in \mathcal{F}_i \quad (17)$$

for edges $(i, j) \in T$. The mutual information $I_{ij}^{\mathcal{D}}$ between facets i and j is given as

$$I_{ij}^{\mathcal{D}} = \sum_{a \in \mathcal{F}_i} \sum_{b \in \mathcal{F}_j} \Theta_{ij}^*(a, b) \log \frac{\Theta_{ij}^*(a, b)}{\Theta_i^*(a) \Theta_j^*(b)} \quad (18)$$

The optimal tree T^* is such that the maximum possible mutual information is encapsulated by its edges.

The algorithm proceeds in two steps, namely, computing the mutual information $I_{ij}^{\mathcal{D}}$ for each pair of facets i and j as in (18); and selecting the top $(K-1)$ edges with the highest mutual information by computing the maximum weight spanning tree for a weighted graph between facets, where

each edge between facet i and j has weight $I_{ij}^{\mathcal{D}}$.

3.4.2 Estimation of Word-Facet Parameters (Λ^*)

A direct computation of the $\log Z(\vec{f}^{(n)})$ is computationally prohibitive. However, using a variational approach and Jensen's inequality, we can compute an approximate estimate $\hat{\Lambda}^*$ characterized by the conditional distribution of query words conditioned on the facets specification

$$\hat{\lambda}_{w;f}^{k*} = \frac{\#(w \in \check{q}, \vec{f}_k = f)}{\#(\vec{f}_k = f)} \quad (19)$$

3.5 User Intent Classification

This section presents a fast online procedure for the user intent classification of a test query as:

$$\hat{f}^\dagger = \arg \max_{\vec{f} \in \mathcal{F}} P(\check{Q} = \check{q}^\dagger | \vec{F} = \vec{f}, \hat{\Lambda}^*) P(\vec{F} = \vec{f} | \Theta_T^*)$$

A naive solution requires computation of $O(\prod_{i \in \mathcal{K}} \mathcal{F}_i)$ product terms, each having $O(L|\mathcal{K}|)$ multiplications. Further, the computation of $Z(\vec{f})$ for each \vec{f} requires summation over \mathcal{V}^m terms, which is computationally infeasible.

Instead, we perform approximate inference using standard Belief Propagation (BP) on a dynamically constructed factor graph based on the model parameters learnt in Section 3.4. The resulting factor graph is tree structured having K variable nodes and $O(MK)$ factor nodes, and the inference procedure converges in at most K iterations.

3.5.1 Dynamic Factor Graph

The factor graph for finding the optimal facet assignment \vec{f}^\dagger for a test query \check{q}^\dagger consists of two classes of factors, namely,

- factors which model the latent relationship between facets in (7); and
- factors which model the impact of query words on facets based on (9) under the simplifying assumption that there are no latent relationships between facets.

Figure 2 shows the factor graph corresponding to the facet classification. The variable nodes, F_1, \dots, F_K , denote the facet variables which can take possible states in $\mathcal{F}_1, \dots, \mathcal{F}_K$, respectively. The factors T_i, T_{ij} model the latent relationship between facets, while the factors W_ℓ^i model the impact of the ℓ^{th} word of the test query \check{q}^\dagger affecting the i^{th} facet variable.

The factors, $T^i, i \in \mathcal{K}$ and $T^{ij}, (i, j) \in T^*$ correspond to the latent relationships between the facets in (7). Their beliefs correspond to the parameters (Θ^*, T^*) based on the training data, given as

$$P_{T^i}(F_i = f) = \Theta_i^*(f)^{(1-d_i)} \quad (20)$$

$$P_{T^{ij}}(F_i = f, F_j = f') = \Theta_{ij}^*(f, f') \quad (21)$$

The factors W_ℓ^i model the conditional distribution $\hat{P}(F_k = f | w \in \check{q})$ under the assumption that all facets are independent of each other. The factors are specified by the beliefs,

$$P_{W_\ell^i}(F_i = f) = \xi_{w;f}^i = \frac{\#(\vec{f}_k = f, w \in \check{q})}{\#(w \in \check{q})} \quad (22)$$

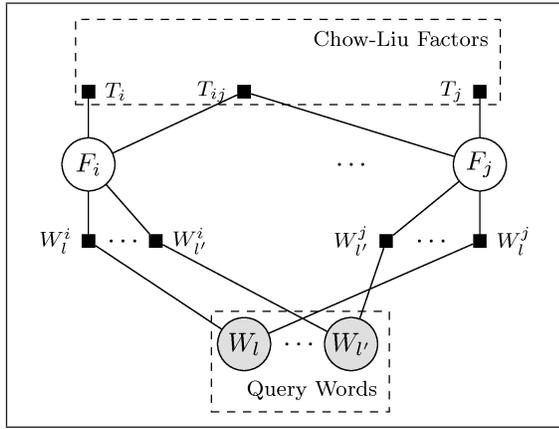


Figure 2: Factor graph for query facet classification. The variable nodes, F_1, \dots, F_K , denote the facets; W_ℓ^i denotes the factor corresponding to the ℓ^{th} query word affecting the i^{th} facet variable; and T_i and T_{ij} are the factors corresponding to the Chow-Liu probability distribution on facet variables which capture the facet interdependencies.

where $\xi_{w,f}^k$ is an approximation for $\hat{P}(F_k = f | w \in \tilde{q})$ obtained as

$$\xi_{w,f}^k \propto P(w \in \tilde{q} | F_k = f, \hat{\Lambda}^*) \hat{P}_1^k(F_k = f) \quad (23)$$

and $\hat{P}_1^k(F_k = f) = \frac{\#(\tilde{f}_k=f)}{N}$ is the MLE of the probability distribution of the k^{th} facet under simplifying assumption that the each facet is independent of other facets.

The parameter $\Xi^* = \{\xi_{w,f}^k : f \in \mathcal{F}_k, k \in \mathcal{K}, w \in \mathcal{V}\}$ models the impact of the words in a query on the facets. Specifically, $\xi_{w,f}^k$ measures the evidence that the k^{th} facet has value f when the query contains the word w .

We note that the evidence $\xi_{w,f}^k$ in (22) is not defined if the word w is not part of the training vocabulary, i.e., the test query \tilde{q}^\dagger contains a word w that is not present in the training data \mathcal{D} . The following section presents a method for incorporating the influence of new words, i.e. words in the test query which are not present in the vocabulary using the semantic relationships between the words and the existing vocabulary.

3.5.2 WordNet Integration

Let w denote a new word not present in the vocabulary $\mathcal{V}_\mathcal{D}$. Then, the corresponding $\xi_{w,f}^i$ are not defined and the previous algorithm cannot be used. However, a semantically related word w' might be present in the vocabulary $\mathcal{V}_\mathcal{D}$; for which the corresponding $\xi_{w',f}^i$ are defined. It is safe to assume that a strong semantic relationship between the words w and w' will be reflected in the resulting facet classifications, reflected in $\xi_{w,f}^i$ and $\xi_{w',f}^i$. Therefore, it is of interest to be able to relate new words in the query to existing words in vocabulary.

We present a two-step method for integrating the capabilities of WordNet [12], a lexical database, into the facet classification for queries containing new words. We first query WordNet for the related word candidates W^\perp with similarity scores $S^\perp = \{s_{w'} : w' \in W^\perp\}$ for a new query word w using Breadth First Search (BFS) till a maximum search depth L . The second step uses the candidates which are present in the vocabulary $W^\star = W^\perp \cap \mathcal{V}_\mathcal{D}$ to compute the parameter

$\tilde{\xi}_{w,f}^i$. Algorithm 2 presents the heuristic procedure for incorporating related words obtained from WordNet to augment results for a new word.

We note that if the word has no neighbours either in WordNet or in the existing vocabulary, the conditional distribution $P(F_i = \cdot | w \in \tilde{q}^\dagger)$ is a Dirichlet distribution with parameter $[\frac{\#(F_i=f_1)}{N}, \dots, \frac{\#(F_i=f_{|\mathcal{F}_i|})}{N}]$. This means that when the test word w has no semantically related neighbours w' with known statistics $\xi_{w',f}^k$, then the conditional distribution is randomly distributed such that, on average, the count statistics for a facet would be same as that found in the training data. However, if there exist neighbours of the word W^\star , that information is incorporated into the conditional distribution for the new word.

Algorithm 2 WordNet integration $\tilde{\xi}_w = F_{\text{WN}}(w)$

Require: L {Maximum depth for BFS}
Require: Ξ^* {Parameter based on \mathcal{D} }
Ensure: $w \notin \mathcal{V}_\mathcal{D}$ {New word not present in vocabulary}
Ensure: $\tau_f^i = \frac{\#(f_i=f)}{N}$
 $(W^\perp, S^\perp) \leftarrow \text{BFS}(w, L)$ {Related words in WordNet}
 $W^\star \leftarrow W^\perp \cap \mathcal{V}_\mathcal{D}$ {Related words in vocabulary}
for $i = 0$ to K **do**
 Initialize $\{\mu_1^i, \dots, \mu_{|\mathcal{F}_i|}^i\} \sim \text{Dir}(\tau_{f_1}^i, \dots, \tau_{f_{|\mathcal{F}_i|}^i})$
 for all f in \mathcal{F}_i **do**
 $\tilde{\xi}_{w,f}^i = \frac{\mu_f^i + \sum_{w' \in W^\star} s_{w'} \xi_{w',f}^i}{\sum_{f \in \mathcal{F}_i} (\mu_f^i + \sum_{w' \in W^\star} s_{w'} \xi_{w',f}^i)}$
 end for
end for
return $\tilde{\xi}_w = \{\tilde{\xi}_{w,f}^i : i \in \mathcal{K}, f \in \mathcal{F}_i\}$

3.5.3 The FastQ Algorithm

Algorithm 3 FastQ $\hat{f}^\dagger = F(\tilde{q}^\dagger | \mathcal{D})$

Require: \tilde{q}^\dagger {Test query}
Require: $\Xi^*, \Theta^*, T^*, \mathcal{V}$ {Parameter from training data \mathcal{D} }
 Initialize $G(\mathcal{K}, \emptyset)$ {factor graph on facet variables}
 for all (i, j) in T^* **do**
 Add factor T^{ij} s.t. $P_{T^{ij}}(F_i = a, F_j = b) = \Theta_{ij}^*(a, b)$
 end for
 for all i in \mathcal{K} **do**
 Add factor T^i s.t. $P_{T^i}(F_i = a) = \frac{1}{\Theta_i^*(a)^{d_a - 1}}$
 end for
 for $l = 1$ to L **do**
 if $q_l^\dagger \in \mathcal{V}$ **then**
 for all i in \mathcal{K} **do**
 Add factor W_ℓ^i s.t. $P_{W_\ell^i}(F_i = f) = \xi_{q_l^\dagger, f}^i \forall f \in \mathcal{F}_i$
 end for
 else
 $\tilde{\xi}_{q_l^\dagger} = F_{\text{WN}}(q_l^\dagger)$ {Use WordNet for new query word}
 for all i in \mathcal{K} **do**
 Add factor W_ℓ^i s.t. $P_{W_\ell^i}(F_i = f) = \tilde{\xi}_{q_l^\dagger, f}^i \forall f \in \mathcal{F}_i$
 end for
 end if
 end for
 $\hat{f}^\dagger = \text{MAX-PROD}(G)$ {Standard BP}
return $\hat{f}^\dagger = \{f_1^\dagger, \dots, f_K^\dagger\}$

The overall procedure for facet classification when a new query, $\hat{q}^\dagger = \{q_1, \dots, q_L\}$ is given in Algorithm 3. The algorithm constructs the dynamic factor graph by adding the factors T_i, T_{ij} corresponding to the probability distribution on facets constructed using the Chow-Liu algorithm. Then, if a query word q_ℓ is in the vocabulary \mathcal{V} , it adds words to the corresponding word factors, and constructs the word factor W_ℓ^i based on parameter Ξ^* , as in (22). Otherwise, if query word q_ℓ is not in vocabulary \mathcal{V} , it constructs the factors W_ℓ^i based on semantically related words in vocabulary \mathcal{V} found using WordNet as in Algorithm 2. The facet assignment \hat{f}^\dagger for the test query is computed using a standard implementation of the Max-Product Belief Propagation [18].

CLAIM 1. *The FastQ algorithm has the following properties*

- (a) *The dynamic factor graph is a tree consisting of K variable nodes and $O(MK)$ factor nodes,*
- (b) *The Max-Product Belief Propagation (BP) converges in at most K iterations,*
- (c) *The average time for query classification is $O(K + \bar{c}_{wn}^L)$ where L is the maximum search depth and \bar{c}_{wn} is the average number of semantically related words for a word in WordNet.*

PROOF. (a) By construction.

- (b) BP converges on trees iterating in at most the number of variable nodes in the tree [18, 30].
- (c) The average time spent in WordNet search is $O(\bar{c}_{wn}^L)$ and the BP algorithm takes $O(K)$ time. \square

The small size of the constructed factor graph allows an implementation with limited resources. The maximum search depth L in WordNet is a user-controlled parameter which can be reduced to speedup the classification.

3.6 Generalizing the Query Intent Model

This section discusses the procedure for generalizing our model in order to explicitly study some other aspect of user intent. For example, the facet *Medical Sensitive* might be facet of interest relevant to medical search engines such as PubMed⁴.

Our query intent model and inference procedure is generic and can be used with other multi-dimensional descriptions of user intent. The requirement is the use of training data which has been classified according to the facet description of interest. Thus, in order to add the facet *Medical Sensitive*, one has to provide the labels for the new facet corresponding to the queries in the training data.

Sometimes, one can avoid the reclassification due to the nature of the data collection procedure. For example, a query having the word *gene* may not be *Medical Sensitive* if the queries are drawn from a generic search engine. However, PubMed queries having the query word *gene* can be assumed to be *Medical Sensitive* (Similar assumptions have been made in [31]). Thus, one can obtain a rough estimate for the normalized influence of words in the query by simply assuming that the queries in PubMed are *Medical Sensitive*, but not in a generic search engine.

⁴<http://www.pubmedcentral.nih.gov>

Dimension	Overall Agreement	Kappa Coefficient
Time S.	99.23	0.98
Scope	96.74	0.93
Objective	92.54	0.85
Authority S.	84.32	0.69
Topic	68.26	0.66
Task	75.71	0.63
Spatial S.	81.07	0.62
Genre	65.00	0.53
Specificity	55.44	0.33

Table 2: Inter-annotation Agreement and Kappa Coefficient for the dimensions of user’s intent.

Alternatively, some of the facets may be known and the remaining might be unknown. For example, if a test query is generated from PubMed, it is safe to assume that the facet *Topic* is *Health* (Similar assumptions have been made in [31]). It is easy to condition on some facets if they are known. The inference for the remaining facets is conditioned [11] over the facet *Topic* having state *Health*.

4. EXPERIMENTAL SETTING

4.1 Dataset

The evaluation data consists of 5,249 queries from a vertical search engine query log in a Spanish speaking country. The sampling of queries took place randomly after calculating the popularity of each query (i.e. the number of times a query was posed), to capture the Zipf’s law distribution (in [2] only popular queries were considered). Hence, 15% were taken from to the set of the most popular queries, 15% from the (long) tail set, and the remaining 70% was sampled from the middle set (i.e., queries with average popularity).

The complete collection of queries was manually classified, assigning a value for each of the dimensions described in Section 3.1. Additionally, 10% of the queries (randomly selected) were classified by two judges. In order to measure the consistency of the judges two well known metrics were applied: the Overall Agreement [13] and Kappa Coefficient [25]. The former metric express a score of how much homogeneity, or consensus, there is in the ratings given by judges; however it does not take into account the agreement that would have been expected due solely to chance. In order to overcome this problem, we used the second metric (i.e., the Kappa Coefficient), which is the proportion of agreement between judges after chance agreement is removed from consideration. Table 2 contains the overall agreement of the judges as well as the Kappa coefficient values.

Results from the overall agreement indicate that the consistency of the manual classification is highly satisfactory. In average, the overall agreement is 80%. Eight out of the nine dimensions have reached an overall agreement higher than 65%, which is quite high if we consider the number of dimensions that were assessed, the number of possible values that each dimension can take, as well as the different criteria and the subjectivity of the judges.

In terms of the Kappa coefficient the values obtained reflect a consistency and reliability of the manual classification. The assessment from the judges for all the dimensions is beyond chance, and more over, for three dimensions the agreement was almost perfect.

Training and testing sets. The amount of labelled

% Tr. data	1%			5%			10%			50%		
	BASE	WN	FASTQ	BASE	WN	FASTQ	BASE	WN	FASTQ	BASE	WN	FASTQ
author s.	33.48	43.44	79.85	42.35	55.41	79.76	50.07	61.64	81.14	55.97	64.82	81.70
genre	30.73	35.51	51.95	38.53	43.85	59.76	43.18	47.42	60.63	50.00	52.47	65.02
objective	81.96	80.03	73.09	84.99	82.51	83.35	86.03	83.24	83.01	86.19	85.62	85.59
scope	23.51	45.42	97.56	37.88	60.75	97.51	46.69	67.67	97.40	56.51	72.79	97.28
spatial s.	45.42	45.15	59.42	49.01	49.68	61.62	52.82	54.39	62.14	56.27	58.32	67.98
specificity	75.62	73.55	77.51	75.93	74.37	76.98	77.24	73.96	77.51	77.08	76.41	78.14
task	68.47	66.61	61.45	72.60	69.16	71.02	75.21	71.90	71.64	76.87	75.78	76.04
time s.	21.25	42.54	97.98	35.55	59.32	98.60	44.61	64.87	98.64	54.37	71.38	98.43
topic	11.73	12.85	14.01	21.97	20.75	19.39	27.57	26.90	25.67	34.44	36.00	32.90

Table 3: This table compares the classification accuracy (in %) of the user intent dimensions for the different models. The mean accuracies for $N = 10$ trials are shown where fixed number (% tr data) of randomly chosen queries were used for training and the remaining data used as test queries. The highest classification accuracy obtained for each facet at given amount of training data (% tr data) is highlighted.

data available in most IR scenarios is much smaller than the corresponding test sets. In order to test the models in the low data regime, we have used 1, 5, 10, 50 % of the available queries as training data. We have repeated the experiment for $N = 10$ independent trials where the training samples have been randomly chosen. We report the resulting mean and variance across the trials.

4.2 Models Tested

The two key components of *FastQ* are the modeling of the latent relationships between facets and the incorporation of WordNet lexical database. In order to highlight the impact of the above two factors in the overall results and establish a baseline, we construct two variants of the *FastQ* algorithm described below:

- *BASE*: The model ignores the latent relationships between facets and does not incorporate WordNet to compute the influence of new words in the test query. Thus, the resulting factor graph does not have the factors T_i and T_{ij} ; and only has factors W_ℓ^i for the words which are already present in the vocabulary \mathcal{V} ; ignoring the test query words not present in the vocabulary.
- *WN*: This does not model the latent relationships between facets; but does incorporate WordNet to incorporate the influence of new words in the test query in terms of their semantic neighbours in the existing vocabulary. Thus, the resulting factor graph does not have the factors T_i and T_{ij} ; but, has factors, W_ℓ^i for the new words in the test query in addition to factors for known words.

4.3 Results

4.3.1 Classification Accuracy

Table 3 presents the classification accuracy of the individual facets. We note that the latent modeling of the facets in *FastQ* improves the accuracy compared to *BASE* and *WN* for the facets: Time Sensitivity, Scope, Spatial Sensitivity, Authority Sensitivity, Genre.

We use the Hamming error to measure the overall accuracy of the prediction, i.e., for a query q^\dagger with true facet classification f^\dagger and the algorithm prediction \hat{f} ; the Hamming error $d_H(f^\dagger, \hat{f})$ is $d_H(f^\dagger, \hat{f}) = \sum_{k=1}^K \mathbf{1}_{\hat{f}_k \neq f_k^\dagger}$ where $\mathbf{1}_{\hat{f}_k \neq f_k^\dagger}$ is 1 if the k^{th} facets are not same; else 0. A low (ideally zero)

	True	Pred.	P	R	F1
Inf	1382	1584	0.79	0.91	0.85
NotInf	645	635	0.73	0.72	0.72
Both	212	20	0.89	0.08	0.15

Table 4: This table shows the classification (*FastQ*) results for the facet *Task* as Informational (*Inf*), Not Informational (*NotInf*) and Both (*Both*). The table shows precision (*P*), recall (*R*) and the F1-score (*F1*) when 50% of the data is used for training and the rest as test queries.

Hamming distance means few (ideally *none*) of the K facets have been incorrectly predicted.

Figure 3(a)-3(c) shows the cumulative distribution of the Hamming errors (number of individually incorrect facets) in the prediction of K facets for 1%, 10% and 50% training data. For example, when 10% of the data is used for training, *FastQ* classifies over 60% of the test queries with at most two mis-classified facets; compared to *WN* (45%) and *BASE* (40%). The modeling of the facet inter-relationships (*FastQ*) improves the overall quality of classification compared to *BASE* and *WN*, especially at low amounts of training data.

We note that *FastQ* performance compared to *BASE* is slightly inferior on the facet “Topic”, which has 23 possible states. Therefore, accurately modelling the probability distribution requires more data. We note that at 90% training data⁵, *FastQ* does perform better on “Topic” (and all other facets) compared to *BASE*. We note that in a real world setting, the amount of available data might be greater than a few thousand queries (for example, when using ODP categories). Under such scenario, *FastQ* will perform better modeling compared to the results shown here.

We attempt to learn a multi-faceted classification of user intent, which is inherently more sophisticated compared to learning the classical notion of user intent as *informational*, *navigational* or *transactional*. However, the closest analogue in our case is the *Task* dimensions whose values are *informational*, *not information* or *both*.

Table 4 shows the classification results for the *Task* dimension when 50% of the queries was used as training data, and the remaining for testing. An average accuracy of 76% is achieved by *FastQ* in the classification of *Task*.

⁵not shown here due to space restriction

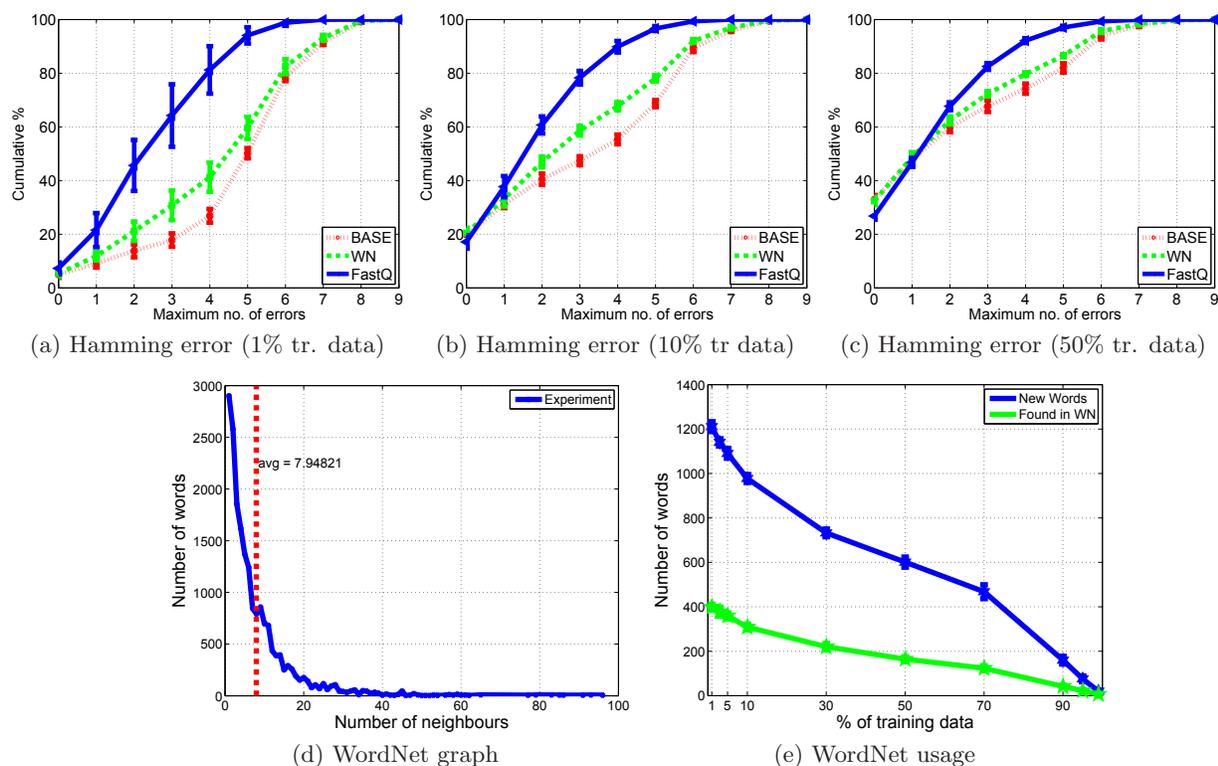


Figure 3: (a)-(c) The mean and variance of the Hamming error for the overall facet classification for the models *BASE* (red), *WN* (green) and *FastQ* (blue). (d) The histogram showing the number of related words obtained using WordNet for a test query word (blue); and the expected (average) number of related words for a new query word (red). (e) This figure shows the mean and variance of the number of new words found in the test data (blue); and the number of words for which related words were found in the training data using WordNet (green); with increasing amount of training data. All figures are based on results obtained from $N = 10$ independent trials.

Method	Size	Accuracy
Topic/homepage [17] [‡]	200	91%
<i>FastQ</i> (<i>Task</i>)	5249	76%
Rule-based hierarchy [15]	1500000	74%
Query-log based [19]	50	54%
Manual classification [2]	6042	50%

Table 5: Comparison of existing approaches for intent classification with size of the dataset and the classification accuracy. ([‡]Results of [17] are based on TREC test collection.)

Table 5 shows *FastQ* compares favorably to existing methods. We re-iterate the multi-dimensional nature of our work, i.e. *Task* is one of the nine facets which characterize user intent in our case, compared to past approaches.

4.3.2 Impact of WordNet

A comparison of the results in Table 3 and Figure 3(a)-3(c) show the incorporation of WordNet for computing the influence of new words in test query based on semantic neighbours in the existing vocabulary improves the quality of results in *WN* compared to *BASE*, especially at low amounts of training data (1 – 10%).

Figure 3(e) shows that around 33% of the new words are semantically related to known words in the training data. This might be surprising, but this is explained from the

characteristics of the training set that only includes 15% of queries from the head of the query distribution. Thus, incorporating WordNet may allow a better modeling for a significant portion of unseen words in a practical web search scenario.

Figure 3(d) shows the histogram of the number of related words in WordNet, experimentally observed, for a query word that is not present in training data, and its average $\bar{c}_{wn} \simeq 8$. This affects the time spent in search for related candidates of a word not present in training data.

The experiments reported used a breadth first search till maximum depth 3 using the words falling in the *synsets* category. We observed that the similarity scores for the neighbours often is either very close to one, or slightly above zero.

4.3.3 Efficiency and Scalability

The average time for query classification at WordNet maximum search depth 3 was observed to be 2–3 ms on an AMD Turion64 machine with 4 GB RAM, which is considerably less than the average latency for a search query which is 200 ms [8]. This allows the incorporation of *FastQ* as input to the ranking algorithm of a search engine; using the user intent information to improve the user experience.

In addition, our results show that with a small training set (a few thousand queries), we can get good predictive

performance. Hence, with training sets available for large search engines the results might be quite better.

5. CONCLUSIONS AND FUTURE WORK

We have presented an efficient and convergent algorithm for inferring the intent of the web search query, described in terms of multiple facets. Experimental results show that modelling the latent relationships between facets improves the accuracy of prediction. Further, the incorporation of WordNet improves intent classification using the semantic relationships in the language vocabulary.

One possible area of improvement is the usage of multiple semantic relationships, e.g. *synonyms*, *antonyms*, *hyponyms*, available in WordNet, which might improve the efficacy of WordNet in intent classification.

Our framework model can be used for fast contextualized ranking of results of a user query with or without click-through data. For example, it can be used for a much more refined query classification and clustering. Similarly, the intent classification can be used as input to existing search engine ranking technology [5]. Alternatively, we can use the facet classifications for sorting and displaying the search engine results in a different way creating a new user experience.

In addition, the generality of the framework allows integration of alternative aspects of user intent making it a powerful tool for contextualizing and personalizing search.

6. REFERENCES

- [1] A. Ashkan, C. L. Clarke, E. Agichtein, and Q. Guo. Classifying and characterizing query intent. In *Proc. of ECIR*, 578–586, 2009.
- [2] R. Baeza-Yates, L. Calderón-Benavides, and C. González-Caro. The intention behind web queries. In *Proc. of SPIRE, LNCS* vol. 4209, 98–109, 2006.
- [3] A. Z. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.
- [4] L. Calderón-Benavides, C. González-Caro, and R. Baeza-Yates. Towards a deeper understanding of the user’s query intent. In *Proc. of Query Representation and Understanding Workshop at SIGIR*, 2010.
- [5] O. Chapelle and S. S. Keerthi. Efficient algorithms for ranking with SVMs. *Inf. Retr.*, 13(3):201–215, 2010.
- [6] C. I. Chow, S. Member, and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Inf. Theory*, 14:462–467, 1968.
- [7] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.
- [8] J. Dean. Challenges in building large-scale information retrieval systems. WSDM’09 keynote, 2009.
- [9] D. Downey, S. Dumais, and E. Horvitz. Heads and tails: Studies of web search with common and rare queries. *Proc. of SIGIR*, 847–848, 2007.
- [10] D. Downey, S. Dumais, D. Liebling and E. Horvitz. Understanding the relationship between searchers’ queries and information goals. In *Proc. of CIKM*, 449–458, 2008.
- [11] F. Eaton and Z. Ghahramani. Choosing a variable to clamp: Approximate inference using conditioned belief propagation. In *Proc. of AISTATS*, 2009.
- [12] P.-V. Eds. *EuroWordNet. A multilingual database with lexical semantic networks*. Kluwer Academic, 1998.
- [13] G. Hripcsak and A. S. Rothschild. Agreement, the f-measure, and reliability in information retrieval. *JAMIA*, 12(3):296–298, May–Jun 2005.
- [14] J. Hu, G. Wang, F. Lochovsky, J. T. Sun, and Z. Chen. Understanding user’s query intent with wikipedia. In *Proc. of WWW*, 471–480, 2009.
- [15] B. J. Jansen, D. L. Booth, and A. Spink. Determining the informational, navigational, and transactional intent of web queries. *Inf. Process. Manage.*, 44(3):1251–1266, 2008.
- [16] B. J. Jansen, A. Spink, and I. Taksa. *Handbook of Research on Web Log Analysis*. Idea Group Inc., 2008.
- [17] I.-H. Kang and G. Kim. Query type classification for web document retrieval. In *Proc. of SIGIR*, 64–71, 2003. ACM.
- [18] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. on Inf. Theory*, 47:498–519, 1998.
- [19] U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in web search. In *Proc. of WWW*, 391–400, 2005.
- [20] Y. Liu, M. Zhang, L. Ru, and S. Ma. Automatic query type identification based on click through information. *LNCS*, 4182:593–600, 2006.
- [21] M. Meila. An accelerated chow and liu algorithm: Fitting tree distributions to high-dimensional sparse data. In *Proc. of ICML*, 249–257, 1999.
- [22] M. Mendoza and R. Baeza-Yates. A web search analysis considering the intention behind queries. In *Proc. of LA-WEB*, 66–74, 2008.
- [23] V. B. Nguyen and M.-Y. Kan. Functional faceted web query analysis. In *Query Log Analysis: Social And Technological Challenges. A workshop at the WWW’07*, May 2007.
- [24] E. Pitler and K. Church. Using word-sense disambiguation methods to classify web queries by intent. In *Proc. of EMNLP*, Jan 2009.
- [25] J. J. Randolph. Free-marginal multirater kappa: An alternative to fleiss fixed-marginal multirater kappa. Joensuu Learning and Instruction Symposium, October 2005.
- [26] D. E. Rose and D. Levinson. Understanding user goals in web search. In *Proc. of WWW*, 13–19, 2004.
- [27] V. Y. F. Tan, A. Anandkumar, L. Tong, and A. S. Willsky. A large-deviation analysis for the maximum likelihood learning of tree structures. In *Proc. of ISIT*, 1140–1144, 2009.
- [28] J. Teevan, S. Dumais, and E. Horvitz. Potential for personalization. *ACM Trans. on Computer*, Jan 2010.
- [29] J. Teevan, S. Dumais, and D. Liebling. To personalize or not to personalize: modeling queries with variation in user intent. In *Proc. of SIGIR*, 163–170, 2008.
- [30] M. J. Wainwright and M. I. Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc., Hanover, MA, USA, 2008.
- [31] R. White, S. Dumais, and J. Teevan. Characterizing the influence of domain expertise on web search behavior. In *Proc. of WSDM*, 132–141, 2009.
- [32] Web query classification. http://en.wikipedia.org/wiki/Web_query_classification.