

# A Joint Information Model for $n$ -best Ranking

**Patrick Pantel**

Yahoo! Inc.  
Santa Clara, CA 95054  
me@patrickpantel.com

**Vishnu Vyas**

USC Information Sciences Institute  
Marina del Rey, CA  
vishnu@isi.edu

## Abstract

In this paper, we present a method for modeling joint information when generating  $n$ -best lists. We apply the method to a novel task of *characterizing the similarity of a group of terms* where only a small set of many possible semantic properties may be displayed to a user. We demonstrate that considering the results jointly, by accounting for the information overlap between results, generates better  $n$ -best lists than considering them independently. We propose an information theoretic objective function for modeling the joint information in an  $n$ -best list and show empirical evidence that humans prefer the result sets produced by our joint model. Our results show with 95% confidence that the  $n$ -best lists generated by our joint ranking model are significantly different from a baseline independent model  $50.0\% \pm 3.1\%$  of the time, out of which they are preferred  $76.6\% \pm 5.2\%$  of the time.

## 1 Introduction

Ranking result sets is a pervasive problem in the NLP and IR communities, exemplified by keyword search engines such as Google (Brin and Page 1998), machine translation systems (Zhang et al. 2006), and recommender systems (Shardanand and Maes 1995; Resnick and Varian 1997).

Consider the lexical semantics task of *explaining why a set of terms are similar*: given a set of terms and a large set of possible explanations for

their similarity, one must choose only the best  $n$  explanations to display to a user. There are many ways to explain why terms are similar<sup>2</sup>; one way is to list the semantic properties that are shared by the terms. For example, consider the following set of terms corresponding to *fruit names*:

{apple, ume, pawpaw, quince}

Example semantic properties that could be used to explain their similarity include: *they are products, they can be eaten, they are solid* (but not *they are companies*, for example). The list of such semantic properties can be very large and some are much more informative than others. For example, the property *can-be-eaten* is much more informative of the similarity of {apple, ume, pawpaw, quince} than the property *is-solid*. Using a simple measure of association between properties and queries, explained in detail later in this paper, one can rank each property and obtain the following three highest scoring properties for explaining the similarity of these terms:

{they are products, they can be imported, they can be exported}

Even though *can be imported* and *can be exported* are highly ranked explanations, taken jointly, once we know one the other does not offer much more information since most things that can be imported can also be exported. In other words, there is a large overlap in information between the two properties. A more informative set of explanations could be obtained by replacing one of these two properties with a property that scored lower but had less information overlap with the others, for example:

---

© 2008. Licensed under the *Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported* license (<http://creativecommons.org/licenses/by-nc-sa/3.0/>). Some rights reserved.

---

<sup>2</sup> In (Vyas and Pantel 2008), we explore the task of explaining the similarity between terms in detail. In this paper, we focus on the task of choosing the best set of explanations given a set of candidates.

{they are products, they can be imported, they can be eaten}

Even though, taken alone, the property *can be eaten* may not be as informative as *can be exported*, it does indeed add more information to the explanation set when considered jointly with the other explanations.

In this paper, we propose an information theoretic objective function for modeling the joint information in an  $n$ -best list. Derived using conditional self-information, we measure the amount of information that each property contributes to a query. Intuitively, when adding a new property to a result set, we should prefer a property that contributes the maximum amount of information to the existing set. In our experiments, we show empirical evidence that humans prefer our joint model's result sets on the task of explaining why a set of terms are similar.

The remainder of this paper is organized as follows. In the next section, we review related literature and position our contribution within that landscape. Section 3 presents the task of explaining the similarity of a set of terms and describes a method for generating candidate explanations from which we will apply our ranking model. In Section 4, we formally define our ranking task and present our Joint Information Ranking model. Experimental results are presented in Section 5 and finally, we conclude with a discussion and future work.

## 2 Related Work

There are a vast number of applications of ranking and its importance to the commercial success at companies such as Google and Yahoo have fueled a great deal of research in recent years. In this paper, we investigate one particular aspect of ranking, the importance of considering the results in an  $n$ -best list jointly because of the information overlap issues described in the introduction, and one particular application, namely explaining why a set of terms are similar.

Considering results jointly is not a new idea and is very similar to the concept of diversity-based ranking introduced in the IR community by Carbonell and Goldstein (1998). In short, selecting an  $n$ -best list is a balancing act between maximizing the relevance of the list and the information novelty of its results. One commonly used approach is to define a measure of novelty/semantic similarity between documents and to apply heuristics to reduce the relevance score of a result item (a hit) by a function of the similarity

of this item to other results in the list (Carbonell and Goldstein 1998; Zhu et al. 2007). Another common approach is to cluster result documents according to their semantic similarity and present clusters to users instead of individual documents (Hearst and Pedersen 1996; Leuski 2001; Liu and Croft 2004). In this paper, we argue that the balance between relevance and novelty can be captured by a formal model that maximizes the joint information content of a result set. Instead of ranking documents in an IR setting, we focus in this paper on a new task of selecting the best semantic properties that describe the similarity of a set of query terms.

By no means an exhaustive list, the most commonly cited ranking and scoring algorithms are HITS (Kleinberg 1998) and PageRank (Page et al. 1998), which rank hyperlinked documents using the concepts of hubs and authorities. The most well-known keyword scoring methods within the IR community are the *tf-idf* (Salton and McGill 1983) and pointwise mutual information (Church and Hanks 1989) measures, which put more importance on matching keywords that occur frequently in a document relative to the total number of documents that contain the keyword (by normalizing term frequencies with inverse document frequencies). Various methods including *tf-idf* have been comparatively evaluated by Salton and Buckley (1987). Creating  $n$ -best lists using the above algorithms produce result sets where each result is considered independently. In this paper, we investigate the utility of considering the result sets jointly and compare our joint method to a pointwise mutual information model.

Within the NLP community,  $n$ -best list ranking has been looked at carefully in parsing, extractive summarization (Barzilay et al. 1999; Hovy and Lin 1998), and machine translation (Zhang et al. 2006), to name a few. The problem of learning to rank a set of objects by combining a given collection of ranking functions using boosting techniques is investigated in (Freund et al. 2003). This rank boosting technique has been used in re-ranking parsers (Collins and Koo 2000; Charniak and Johnson 2005). Such re-ranking approaches usually improve the likelihood of candidate results using extraneous features and, for example in parsing, the properties of the trees. In this paper, we focus on a difference task: the lexical semantics task of selecting the best semantic properties that help explain why a set of query terms are similar. Unlike in parsing and machine translation, we are not ulti-

mately looking for the best single result, but instead the  $n$ -best.

Looking at commercial applications, there are many examples showcasing the importance of ranking, for example Internet search engines like Google and Yahoo (Brin and Page 1998). Another application is online recommendation systems where suggestions must be ranked before being presented to a user (Shardanand and Maes 1995). Also, in online social networks such as Facebook and LinkedIn, new connections or communities are suggested to users by leveraging their social connections (Spretus, et al. 2005).

### 3 Explaining Similarity

Several applications, such as IR engines, return the  $n$ -best ranked results to a query. Although we expect our joint information model, presented in Section 4.2, to generalize to many ranking tasks, our focus in this paper is on the task of choosing the  $n$ -best explanations that describe the similarity of a set of terms. That is, given a set of terms, one must choose the best set of characterizations of why the terms are similar, chosen from a large set of possible explanations.

Analyzing the different ways in which one can explain/characterize the similarity between terms is beyond the scope of this paper<sup>3</sup>. The types of explanations that we consider in this paper are semantic properties that are shared by the terms. For example, consider the query terms  $\{apple, ume, pawpaw, quince\}$  presented in Section 1. An example set of properties that explains the similarity of these words might include  $\{they\ are\ products, they\ can\ be\ imported, they\ can\ be\ exported, they\ are\ tasty, they\ grow\}$ .

The range of possible semantic properties is large. For the above example, we may have offered many other properties like  $\{they\ are\ entities, they\ can\ be\ eaten, they\ have\ skin, they\ are\ words, they\ can\ be\ roasted, they\ can\ be\ shipped, etc.\}$  Choosing a high quality concise set of properties is the goal of this paper.

Our hypothesis is that considering items in a result set jointly for ranking produces better result sets than considering them independently. An important question then is: what is a utility function for measuring a *better* result? We propose that a result set is considered better than another if a person could more easily reconstruct the original query from it. Or, in other words, a result set is considered better than another if it

reduces more the uncertainty of what the original query was. Here, reducing the uncertainty means making it easier for a human to understand the original question (i.e., a good explanation should clarify the query).

Formally, we define our ranking task as:

**Task Definition:** Given a query  $Q = \{q_1, q_2, \dots, q_m\}$  and a set of candidate properties  $R = \{r_1, r_2, \dots, r_k\}$ , where  $q$  is a term and  $r$  is a property, find the set of properties  $R' = \{r_1, r_2, \dots, r_n\}$  that most reduces the uncertainty of  $Q$ , where  $n \ll k$ .

Recall from Section 1 the example  $Q = \{apple, ume, pawpaw, quince\}$ . The set of properties:

$\{they\ are\ products, they\ can\ be\ imported, they\ can\ be\ eaten\}$

is preferred over the set

$\{they\ are\ products, they\ can\ be\ imported, they\ can\ be\ exported\}$

since it reduces more the uncertainty of what the original query is. That is, if we hid the query  $\{apple, ume, pawpaw, quince\}$  from a person, the first set of properties would help more that person guess the query elements than the second properties.

In Section 4, we describe two models for measuring this uncertainty reduction and in Section 5.1, we describe an evaluation methodology for quantifying this reduction in uncertainty using human judgments.

#### 3.1 Source of Properties

What is the source of the semantic properties to be used as explanations? Following Lin (1998), we use syntactic dependencies between words to model their semantic properties. The assumption here is that some grammatical relations, such as *subject* and *object* can often yield semantic properties of terms. For example, given enough corpus occurrences of a phrase like “*students eat many apples*”, then we can infer the properties *can-be-eaten* for *apples* and *can-eat* for *students*. Unfortunately, many grammatical relations do not specify semantic properties, such as most *conjunction* relations for example. In this paper, we use a combination of corpus statistics and manual filters of grammatical relations (such as omitting conjunction relations) to uncover candidate semantic properties, as described in the next section. With this method, we unfortunately uncover some non-semantic properties and fail to uncover some correct semantic properties.

<sup>3</sup> This topic is the focus of (Vyas and Pantel 2008).

Improving the candidate lists of semantic properties is grounds for further investigation.

### 3.2 Extracting Properties

Given a set of similar terms, we look at the overlapping syntactic dependencies between the words in the set to form candidate semantic properties. Example properties extracted by our system (described below) for a random sample of two instances from a cluster of food,  $\{apple, beef\}$ , include<sup>4</sup>:

```
shredded, sliced, lean, sour, de-
licious, cooked, import, export,
eat, cook, dice, taste, market,
consume, slice, ...
```

We obtain candidate properties by parsing a large textual corpus with the Minipar parser (Lin 1993)<sup>5</sup>. For each word in the corpus, we extract all of its dependency links, forming a feature vector of syntactic dependencies. For example, below is a sample of the feature vector for the word *apple*:

```
adj-mod:gala, adj-mod:shredded,
object-of:caramelize, object-of:eat,
object-of:import, ...
```

Intersecting *apple*'s feature vector with *beef*'s, we are left with the following candidate properties:

```
adj-mod:shredded, object-of:eat,
object-of:import, ...
```

In this paper, we omit the relation name of the syntactic dependencies, and instead write:

```
shredded, eat, import, ...
```

This list of syntactic dependencies forms the candidate properties for our ranking task defined in Section 3.

In Section 4, we use corpus statistics over these syntactic dependencies to find the most informative properties that explain the similarity of a set of terms. Some syntactic dependencies are not reliably descriptive of the similarity of words such as conjunctions and determiners. We omit these dependency links from our model.

## 4 Ranking Models

In this section, we present our ranking models for choosing the  $n$ -best results to a query according to our task definition from Section 3. The models

are expected to generalize to many ranking tasks, however in this paper we focus solely on the problem of choosing the best semantic properties that describe the similarity of a set of terms.

In the next section, we outline our baseline independent model, which is based on a commonly used ranking metric in lexical semantics for selecting the most informative properties of a term. Then in Section 4.2, we propose our new model for considering the properties jointly.

### 4.1 EIIR: Expected Independent Information Ranking Model (Baseline Model)

Recall the task definition from Section 3. Finding a property  $r$  that most reduces the uncertainty in a query set  $Q$  can be modeled by measuring the strength of association between  $r$  and  $Q$ . Following Pantel and Lin (2002), we use pointwise mutual information (*pmi*) to measure the association strength between two events  $q$  and  $r$ , where  $q$  is a term in  $Q$  and  $r$  is syntactic dependency, as follows (Church and Hanks 1989):

$$pmi(q, r) = \log \frac{\frac{c(q, r)}{N}}{\frac{\sum_{w \in W} c(w, r)}{N} \times \frac{\sum_{f \in F} c(q, f)}{N}} \quad (4.1)$$

where  $c(q, r)$  is the frequency of  $r$  in the feature vector of  $q$  (as defined in Section 3.2),  $W$  is the set of all words in our corpus,  $F$  is the set of all syntactic dependencies in our corpus, and

$N = \sum_{w \in W} \sum_{f \in F} c(w, f)$  is the total frequency count of all features of all words.

We estimate the association strength between a property  $r$  and a set of terms  $Q$  by taking the expected *pmi* between  $r$  and each term in  $Q$  as:

$$pmi(Q, r) = \sum_{q \in Q} P(q) pmi(q, r) \quad (4.2)$$

where  $P(q)$  is the probability of  $q$  in the corpus.

Finally, the EIIR model chooses an  $n$ -best list by selecting the  $n$  properties from  $R$  that have highest  $pmi(Q, r)$ .

### 4.2 JIR: Joint Information Ranking Model

The hypothesis of this paper is that considering items in an  $n$ -best result set jointly for ranking produces better result sets than considering them independently, an example of which is shown in Section 1.

Recall our task definition from Section 3: to select an  $n$ -best list  $R'$  from  $R$  such that it most reduces the uncertainty of  $Q$ . Recall that for explaining the similarity of terms,  $Q$  is the set of

<sup>4</sup> We omit the syntactic relations for readability.

<sup>5</sup> Section 5.1 describes the specific corpus and method that was used to obtain our reported results.

query words to be explained and  $R$  is the set of all properties shared by words in  $Q$ . The above task of finding  $R'$  can be captured by the following objective function:

$$R' = \arg \min_{R' \subset R} I(Q|R') \quad (4.3)$$

where  $I(Q|R')$  is the amount of information in  $Q$  given  $R'$ .<sup>6</sup>

$$I(Q|R') = \sum_{q \in Q} P(q) \times I(q|R') \quad (4.4)$$

where  $P(q)$  is the probability of term  $q$  in our corpus (defined in the Section 4.1) and  $I(q|R')$  is the amount of information in  $q$  given  $R'$ , which is defined as the conditional self-information between  $q$  and  $R'$  (Merhav and Feder 1998):

$$\begin{aligned} I(q|R') &= I(q|r_1, r_2, \dots, r_n) \\ &= -\log P(q|r_1, r_2, \dots, r_n) \\ &= -\log \frac{c(q, R')}{c(*, R')} \end{aligned} \quad (4.5)$$

where  $c(q, R')$  is the frequency of all properties in  $R'$  occurring with word  $q$  and  $*$  represents all possible terms in the corpus<sup>7</sup>. We have:

$$c(q, R') = \sum_{r \in R'} c(q, r) \quad \text{and} \quad c(*, R') = \sum_{r \in R'} \sum_{q \in Q'} c(q, r)$$

where  $c(q, r)$  is defined as in Section 4.1 and  $Q'$  is the set of all words that have all the properties in  $R'$ . Computing  $c(*, R')$  efficiently can be done using a reverse index from properties to terms.

The *Joint Information Ranking* model (JIR) is the objective function in Eq. 4.3. We find a sub-optimal solution to Eq. 4.3 using a greedy algorithm by starting with an empty set  $R'$  and iteratively adding one property  $r$  at a time into  $R'$  such that:

$$r = \arg \min_{r \in R - R'} \sum_{q \in Q} P(q) \times I(q|R' \cup r) \quad (4.6)$$

The intuition behind this algorithm is as follows: when choosing a property  $r$  to add to a partial result set, we should choose the  $r$  that contributes the maximum amount of information to the existing set (where all properties are considered jointly).

A brute force optimal solution to Eq. 4.3 involves computing  $I(Q|R')$  for all subsets  $R'$  of size  $n$  of  $R$ . In future work, we will investigate heuristic search algorithms for finding better solutions to Eq. 4.3, but our experimental results discussed in Section 5 show that our greedy solution to Eq. 4.3 already yields significantly better  $n$ -best lists than the baseline EIIR model.

## 5 Experimental Results

In this section, we show empirical evidence that considering items in an  $n$ -best result set jointly for ranking produces better result sets than considering them independently. We validate this claim by testing whether or not human judges prefer the set of explanations generated by our joint model (JIR) over the independent model (EIIR).

### 5.1 Experimental Setup

We trained the probabilities described in Section 4 using corpus statistics extracted from the TREC-9 and TREC-2002 Aquaint collections consisting of approximately 600 million words. We used the Minipar parser (Lin 1993) to analyze each sentence and we collected the frequency counts of the grammatical contexts output by Minipar and used them to compute the probability and pointwise mutual information values from Sections 4.1 and 4.2. Given any set of words  $Q$  from the corpus, our joint and independent models generate a ranked list of  $n$ -best explanations (i.e., properties) for the similarity of the words.

Recall the example set  $Q = \{apple, beef\}$  from Section 3.2. Following Section 3.2, all grammatical contexts output by Minipar that both words share form a candidate explanation set  $R$  for their similarity. For  $\{apple, beef\}$ , our systems found 312 candidate explanations. Applying the independent ranking model, EIIR, we obtain the following top-5 best explanations,  $R'$ :

product, import of, export, ban on, industry

Using the joint model, JIR, we obtain:

export, product, eat, ban on, from menu

### 5.2 Comparing Ranking Models

In order to obtain a representative set of similar terms as queries to our systems, we randomly chose 100 concepts from the CBC collection (Pantel and Lin 2002) consisting of 1628 clusters of nouns. For each of these concepts, we randomly chose a set of cluster instances (nouns),

<sup>6</sup> Note that finding the set  $R'$  that minimizes the amount of information in  $Q$  given  $R'$  equates to finding the  $R'$  that reduces most the uncertainty in  $Q$ .

<sup>7</sup> Note that each property in  $R'$  is shared by  $q$  because of the way the candidate properties in  $R$  were constructed (see Section 3.2).

**Table 1.** Confusion matrix between the two judges on the annotation task over all explanation set sizes ( $n = 1 \dots 5$ ).

	<i>JIR</i>	<i>EIIR</i>	<i>EQUAL</i>
<i>JIR</i>	<b>153</b>	2	48
<i>EIIR</i>	11	<b>33</b>	19
<i>EQUAL</i>	29	<b>7</b>	<b>198</b>

where the size of each set was randomly chosen to consist of two or three noun (chosen to reduce the runtime of our algorithm). For example, three of our randomly sampled concepts were *Music*, *Flowers*, and *Alcohol* and below are the random instances selected from these concepts:

- {concerto, quartet, Fifth Symphony}
- {daffodil, lily}
- {gin, alcohol, rum}

Each of these three samples forms a query. Applying both our *EIIR* and *JIR* models, we generated the top-5 explanations for each of the 100 samples. For example, below are the explanations returned for {daffodil, lily}:

- **EIIR:** *bulb, bouquet of, yellow, pink, hybrid*
- **JIR:** *flowering, bulb, bouquet of, hybrid, yellow*

Two judges then independently annotated 500 test cases using the following scheme. For each of the 100 samples, a judge is presented with the sample along with the top-1 explanation of both systems, randomly ordered for each sample such that the judge can never know which system generated which explanation. The judge then must make one of the following three choices:

- **Explanation 1:** The judge prefers the first explanation to the second.
- **Explanation 2:** The judge prefers the second explanation to the first.
- **Equal:** The judge cannot determine that one explanation is better than the other.

The judge is then presented with the top-2 explanations from each system, then the top-3, top-4, and finally the top-5 explanations, making the above annotation decision each time. Once the judge has seen the top-5 explanations for the sample, the judge moves on to the next sample and repeats this process until all 100 samples are annotated. Allowing the judges to see the top-1, top-2, up to top-5 explanations allows us to later

**Table 2.** Inter-annotator agreement statistics over varying explanation set sizes  $n$ .

$n$	AGREEMENT (%)	KAPPA ( $\kappa$ )
1	75.0	0.47
2	70.0	0.50
3	77.0	0.62
4	78.0	0.63
5	84.0	0.73

inspect how our ranking algorithms perform on different sizes of explanation sets.

The above annotation task was performed independently by two judges and the resulting agreement between the judges, using the Kappa statistic (Siegel and Castellan Jr. 1988), was  $\kappa = 0.60$ . Table 1 lists the full confusion matrix on the annotation task. On just the annotations of the top-5 explanations, the agreement was  $\kappa = 0.73$ . Table 2 lists the Kappas for the different sizes of explanation sets. It is more difficult for judges to determine the quality of smaller explanation sets.

For the above top-5 explanations for the query {daffodil, lily}, both judges preferred the *JIR* properties since *flowering* was deemed more informative than *pink* given that we also know the property *yellow*.

### 5.2.1 Evaluation Results

Table 3 shows sample  $n$ -best lists generated by our system and Table 4 presents the results of the experiment described in the previous section. Table 4 lists the preferences of the judges for the  $n$ -best lists generated by the independent and joint models, in terms of the percentage of samples preferred by each judge on each model. We report our results on both all 500 annotations and on the 100 annotations for the explanation sets of size  $n = 5$ . Instead of using an adjudicator for resolving the two judges’ disagreements, we weighted each judge’s decision by 0.5. We used bootstrap resampling to obtain the 95% confidence intervals.

The judges significantly preferred the joint model over the independent model. Looking at all annotated explanation sets (varying  $n$  from 1 to 5), the  $n$ -best lists from *JIR* were preferred 39.7% of the time. On the  $50.0\% \pm 3.1\%$  test cases where one list was preferred over another, the *JIR* lists were preferred overall  $76.6\% \pm 5.2\%$  of the time, with 95% confidence. Caution should be taken when interpreting the results for  $n < 3$  since the annotator agreement for these was very low. However, as shown in Figure 1, human preference for the *JIR* model was higher at  $n \geq 3$ .

**Table 3.** Five example  $n$ -best lists, drawn from our random sample described in Section 5.1, using the joint *JIR* model and the independent *EIIR* model (for  $n=5$ ).

Query ( $Q$ )	<i>JIR</i> $n$ -best ( $R'$ )	<i>EIIR</i> $n$ -best ( $R'$ )
{gin, alcohol, rum}	drink, spike with, sell, use, consume	sell, drink, use, consume, buy
{Temple University, Michigan State}	political science at, professor at, director at, student at, attend	professor at, professor, director at, student at, student
{concerto, quartet, Fifth Symphony}	Beethoven, his, play, write, performance	his, play, write, performance, perform
{ranch house, loft}	offer, brick, sprawling, rambling, turn-of-the-century	his, live, her, buy, small
{dysentery, tuberculosis}	morbidity, die of, case, patient, suffer from	die of, case, patient, case of, have

**Table 4.** Percentage of test cases where the judges preferred *JIR* vs. *EIIR* vs. they had no preference, computed over all explanation set sizes ( $n = 1 \dots 5$ ) vs. only the explanation sets of size  $n = 5$ .

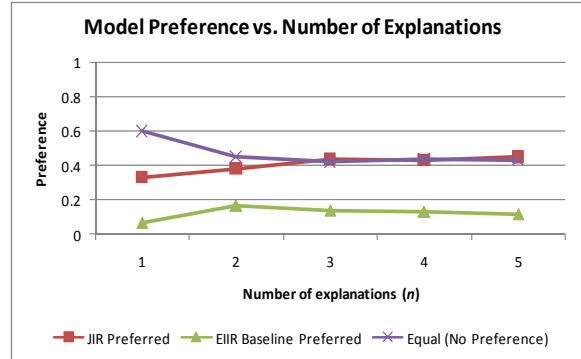
SYSTEM	ALL (95% CONF <sup>†</sup> )	$N=5$ (95% CONF <sup>†</sup> )
<i>JIR</i>	39.7% $\pm$ 3.0%	43.7% $\pm$ 6.9%
<i>EIIR</i>	10.4% $\pm$ 1.3%	10.1% $\pm$ 4.2%
Equal	50.0% $\pm$ 3.1%	45.2% $\pm$ 6.9%

<sup>†</sup>95% confidence intervals estimated using bootstrap resampling.

## 5.2.2 Discussion and Error Analysis

Figure 1 illustrates the annotated preferences over varying sizes of explanation sets, for  $n \in [1 \dots 5]$ . Except in the case where only one explanation is returned, we see consistent preferences between the judges. Manual inspection of the size 1 explanation sets showed that often one property is not enough to understand the similarity of the query words. For example, consider the following two explanation sets: {sell} and {drink}. If you did not know the original query  $Q$ , one list would not be much better than the other in determining what the query was. But, by adding one more property, we get: {sell, drink} and {drink, spike with}. The second explanation list reduces much more the uncertainty that the query consists of *alcoholic beverages*, as you probably guessed (the first list also reduces the uncertainty, but not as much as the second). The above example is taken from our random sample list for the query words {gin, alcohol, rum} – the explanation {drink, spike with} was generated using the *JIR* model.

We manually inspected some of the sample queries where both judges preferred the *EIIR*  $n$ -best list. One such sample query was: {Jerry Falwell, Jim Bakker, Pat Robertson}. The  $n$ -best lists returned by the *JIR* and *EIIR* models respectively were {televangelist, evangelist, Rev., television, founder} and {evangelist, television, Rev., founder, religious}. Both judges preferred the



**Figure 1.** Percentage of human preference for each model with varying sizes of explanation sets ( $n$ ).

*EIIR* list because of the overlap in information between *televangelist* and *evangelist*. The problem here in *JIR* was that the word *televangelist* was very rare in the corpus and thus few terms had both the feature *televangelist* and *evangelist*. We would expect in a larger corpus to see a larger overlap with the two features, in which case *evangelist* would not be chosen by the *JIR* model.

As discussed in Section 2, considering results jointly is not a new idea and is very similar to the concept of diversity-based ranking introduced in the IR community by Carbonell and Goldstein (1998). Their proposed technique, called maximal marginal relevance (MMR), forms the basis of most schemes used today and works as follows. Initially, each result item is scored independently of the others. Then, the  $n$ -best list is selected by iteratively choosing the highest scoring result and then discounting each remaining candidate's score by some function of the similarity (or information gain) between that candidate and the currently selected members of the  $n$ -best list. In practice, these heuristic-based algorithms are fast to compute and are used heavily by commercial IR engines. The purpose of this paper is to investigate a principled definition of diversity using the concept of maximal joint information. The objective function proposed in Eq. 4.3 provides a basis for understanding diversity through the lens of information theory. Although this paper fo-

cuses on the task of explaining the similarity of terms, we plan in future work to apply our method to an IR task in order to compare and contrast our method with MMR.

## 6 Conclusion

This paper investigates the problem of  $n$ -best ranking on the lexical semantics task of explaining/characterizing the similarity of a group of terms where only a small set of many possible semantic properties may be displayed to a user. We propose that considering the results jointly, by accounting for the information overlap between results, helps generate better  $n$ -best lists. We presented an information theoretic objective function, called *Joint Information Ranking*, for modeling the joint information in an  $n$ -best list. On our lexical semantics task, empirical evidence shows that humans significantly prefer JIR  $n$ -best lists over a baseline model that considers the explanations independently. Our results show that the  $n$ -best lists generated by the joint model are judged to be significantly different from those generated by the independent model  $50.0\% \pm 3.1\%$  of the time, out of which they are preferred  $76.6\% \pm 5.2\%$  of the time, with 95% confidence.

In future work, we plan to investigate other joint models using latent semantic analysis techniques, and to investigate heuristic algorithms to both optimize search efficiency and to better approximate our *JIR* objective function. Although applied only to the task of characterizing the similarity of terms, it is our hope that the JIR model will generalize well to many ranking tasks, from keyword search ranking, to recommendation systems, to advertisement placements.

## References

- Barzilay, R.; McKeown, K.; and Elhadad, M. 1999. Information Fusion in the Context of Multi-Document Summarization. In *Proceedings of ACL-1999*. pp. 550-557. College Park, MD.
- Brin, S. and Page, L. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30:107-117.
- Carbonell, J. G. and Goldstein, J. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of SIGIR-1998*. pp. 335-336.
- Charniak, E. and Johnson, M. 2005. Coarse-to-fine  $n$ -best parsing and MaxEnt discriminative reranking. In *Proceedings of ACL-2005*. pp. 173-180. Ann Arbor, MI.
- Church, K. and Hanks, P. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of ACL-89*. pp. 76-83. Vancouver, Canada.
- Collins, M. and Koo, T. 2000. Discriminative Reranking for Natural Language Parsing. In *Proceedings ICML-2000*. pp. 175-182. Palo Alto, CA.
- Freund, Y.; Iyer, R.; Schapier, E.R. and Singer, Y. 2003. An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research*, 4:933-969.
- Harris, Z. 1985. Distributional structure. In: Katz, J. J. (ed.) *The Philosophy of Linguistics*. New York: Oxford University Press. pp. 26-47.
- Hearst, M. A. and Pedersen, J. O. 1996. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of SIGIR-1996*. pp. 76-84. Zurich, Switzerland.
- Hovy, E.H. and Lin, C.-Y. 1998. Automated Text Summarization in SUMMARIST. In M. Maybury and I. Mani (eds), *Advances in Automatic Text Summarization*. Cambridge, MIT Press.
- Kleinberg, J. 1998. Authoritative sources in a hyperlinked environment. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. Pp. 668-677. New York, NY.
- Leuski, A. 2001. Evaluating document clustering for interactive information retrieval. In *Proceedings of CIKM-2001*. pp. 33-40. Atlanta, GA.
- Lin, D. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL-98*. pp. 768-774. Montreal, Canada.
- Lin, D. 1993. Parsing Without OverGeneration. In *Proceedings of ACL-93*. pp. 112-120. Columbus, OH.
- Liu, X. and Croft, W. B. 2004. Cluster-based retrieval using language models. In *Proceedings of SIGIR-2004*. pp. 186-193. Sheffield, UK.
- Merhav, N. and Feder, M. 1998. Universal Prediction. *IEEE Transactions on Information Theory*, 44(6):2124-2147.
- Page, L.; Brin, S.; Motwani R.; Winograd, T. 1998. *The PageRank Citation Ranking: Bringing Order to the Web*. Stanford Digital Library Technologies Project.
- Pantel, P. and Lin, D. 2002. Discovering Word Senses from Text. In *Proceedings of KDD-02*. pp. 613-619. Edmonton, Canada.
- Resnick, P. and Varian, H. R. 1997. Recommender Systems. *Communications of the ACM*, 40(3):56-58.
- Salton, G. and Buckley, C. 1987. Term Weighting Approaches in Automatic Text Retrieval. *Technical Report:TR81-887*, Ithaca, NY.
- Salton, G. and McGill, M. J. 1983. *Introduction to Modern Information Retrieval*. McGraw Hill.
- Shardanand, U. and Maes, P. 1995. Social Information Filtering: Algorithms for Automating "Word of Mouth". In *Proceedings of ACM CHI-1995*. pp. 210-217. New York.
- Siegel, S. and Castellan Jr., N. J. 1988. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill.
- Spretus, E.; Sahami, M.; and Buyukkokten, O. 2005. Evaluating Similarity Measures: A Large-Scale Study in the Orkut Social Network. In *Proceedings of SIGKDD-2005*. pp. 678-684. Chicago, IL.
- Vyas, V. and Pantel, P. 2008. Explaining Similarity of Terms. In *Proceedings of COLING-2008*. Manchester, England.
- Zhu, X.; Goldberg, A.; Van Gael, J.; and Andrzejewski, D. 2007. Improving Diversity in Ranking using Absorbing Random Walks. In *Proceedings of NAACL HLT 2007*. pp. 97-104. Rochester, NY.
- Zhang, Y.; Callan, J.; and Minka, T. 2002. Novelty and redundancy detection in adaptive filtering. In *Proceedings of SIGIR-2002*. pp. 81-88. Tampere, Finland.
- Zhang, Y.; Hildebrand, A. S.; and Vogel, S. 2006. Distributed Language Modeling for  $N$ -best List Re-ranking. In *Proceedings of EMNLP-2006*. Pp. 216-223. Sydney, Australia.