Rank$_{\textbf{Pref}}$: Ranking Sentences Describing Relations between Biomedical Entities with an Application

Catalina O Tudor **K Vijay-Shanker**
Department of Computer and Information Sciences
University of Delaware, Newark, DE, USA
tudor@cis.udel.edu vijay@cis.udel.edu

Abstract

This paper presents a machine learning approach that selects and, more generally, ranks sentences containing clear relations between genes and terms that are related to them. This is treated as a binary classification task, where preference judgments are used to learn how to choose a sentence from a pair of sentences. Features to capture how the relationship is described textually, as well as how central the relationship is in the sentence, are used in the learning process. Simplification of complex sentences into simple structures is also applied for the extraction of the features. We show that such simplification improves the results by up to 13%. We conducted three different evaluations and we found that the system significantly outperforms the baselines.

1 Introduction

Life scientists, doctors and clinicians often search for information relating biological concepts. For example, a doctor might be interested to know the impact of a drug on some disease. One source of information is the knowledge bases and ontologies that are manually curated with facts from scientific articles. However, the curation process is slow and cannot keep up with ongoing publications. Moreover, not all associations between biological concepts can be found in these databases.

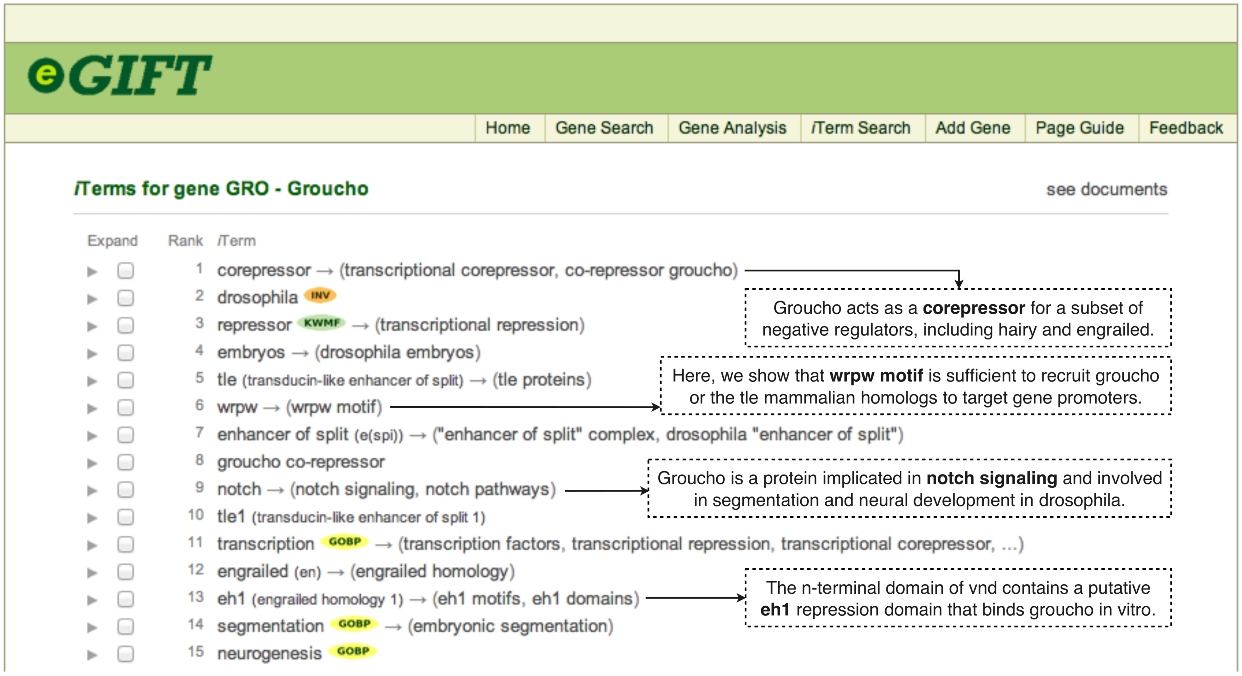
Another source of information is the scientific literature itself. However, searching for biological facts and how they might be related is often cumbersome. The work presented in this paper tries to automate the process of finding sentences that clearly describe relationships between biological concepts. We rank all sentences mentioning two concepts and pick the top one to show to the user. In this paper, we focused on certain specific types of concepts (i.e., genes[1] and terms believed to be related to them), although our approach can be generalized.

Systems to facilitate knowledge exploration of genes are being built for the biomedical domain. One of them, eGIFT (Tudor et al., 2010), tries to identify iTerms (informative terms) for a gene based on frequency of co-occurrence (see Figure 1 for top 15 terms selected for gene *Groucho*). iTerms are unigrams, bigrams, and exact matches of biomedical terms gathered from various controlled vocabularies. Thus, iTerms can be of any type (e.g., processes, domains, drugs, other genes, etc.), the types being determined by what is being described about the gene in the literature. The iTerms for a gene are ranked based on a score that compares their frequencies of occurrence in publications mentioning the gene in question with their frequencies in a background set of articles about a wide variety of genes.

Previous evaluation of eGIFT by life scientists suggested that there is almost always some kind of relationship between a gene and its iTerms. These relationships can be many and varied from one gene-term pair to another. Sometimes a user might make an erroneous assumption about a gene-term association if sentences supporting the association are not immediately inspected. For example, upon seeing "co-repressor" in connection to gene *Groucho*, eGIFT users might correctly assume that *Groucho* is

[1] Throughout the paper, the word "gene" will be used for both the gene and its products.

Proceedings of the 2012 Workshop on Biomedical Natural Language Processing (BioNLP 2012), pages 163–171,
Montréal, Canada, June 8, 2012. ©2012 Association for Computational Linguistics



iTerms for gene GRO - Groucho

Expand	Rank	iTerm
▶ □	1	corepressor → (transcriptional corepressor, co-repressor groucho)
▶ □	2	drosophila INV
▶ □	3	repressor KWMF → (transcriptional repression)
▶ □	4	embryos → (drosophila embryos)
▶ □	5	tle (transducin-like enhancer of split) → (tle proteins)
▶ □	6	wrpw → (wrpw motif)
▶ □	7	enhancer of split (e(spl)) → ("enhancer of split" complex, drosophila "enhancer of split")
▶ □	8	groucho co-repressor
▶ □	9	notch → (notch signaling, notch pathways)
▶ □	10	tle1 (transducin-like enhancer of split 1)
▶ □	11	transcription GOBP → (transcription factors, transcriptional repression, transcriptional corepressor, ...)
▶ □	12	engrailed (en) → (engrailed homology)
▶ □	13	eh1 (engrailed homology 1) → (eh1 motifs, eh1 domains)
▶ □	14	segmentation GOBP → (embryonic segmentation)
▶ □	15	neurogenesis GOBP

Groucho acts as a **corepressor** for a subset of negative regulators, including hairy and engrailed.

Here, we show that **wrpw motif** is sufficient to recruit groucho or the tle mammalian homologs to target gene promoters.

Groucho is a protein implicated in **notch signaling** and involved in segmentation and neural development in drosophila.

The n-terminal domain of vnd contains a putative **eh1** repression domain that binds groucho in vitro.

Figure 1: Top iTerms for gene *Groucho*, and sentences picked by $\mathit{Rank}_{\mathrm{Pref}}$ for various iTerms.

a co-repressor (i.e., a protein that binds to transcription factors). However, upon seeing "wrpw motif", a user might assume that this is a motif contained within gene *Groucho*, as this is typically the association that we make between genes and information annotated for them in knowledge bases. But this would be a wrong assumption, since in actuality the wrpw motif is contained within other genes that interact with *Groucho*, fact which is evident from reading sentences containing the gene and the motif. To get a quick overall understanding of a gene's functionalities, users of eGIFT could be presented with terms extracted for the gene, as well as sentences clearly describing how they are related.

Our method selects sentences using a model that is trained on preference judgments provided by biologists. Example sentences chosen by our method are shown in Figure 1. While we evaluate our approach on sentences from eGIFT, this work could have equally applied on other similar systems (Smalheiser et al., 2008; Gladki et al., 2008; Kim et al., 2008; Kaczanowski et al., 2009). These systems also identify "important terms" from a set of documents retrieved for a given search (either a gene name or other biomedical concept).

The main contributions of this work are: (1) a method for ranking sentences by employing machine learning; (2) the use of preference judgments; (3) features to capture whether two terms are clearly related and in focus in a sentence; (4) another application of sentence simplification, showing a significant gain in performance when utilized.

We continue with a description of our approach, which includes the use of preference judgments to learn the models, how the features are extracted, and how the sentence simplifier is used for this task. The evaluation of the trained model and the system's results are presented in the following section. Related work, conclusions, and future directions are provided at the end of the manuscript.

2 Methods

Rather than pre-judging what is important for this task and manually determining a weighting schema to automatically score sentences for a gene-term pair, we approached this task using machine learning. We asked a group of annotators to rank sentences relating genes and iTerms, and we used their annotations, together with features described in Section 2.3, to learn how to rank sentences.

2.1 Preference Judgments

For the annotation task, we presented biologists with sentences containing a gene-term pair and asked them to specify which sentence they prefer. One way to do this is by employing the pointwise approach, which requires absolute judgments (i.e. the annotator scores each sentence in a list or ranks the sentences based on their relevance to the given task). A second approach is the pairwise approach, which requires the iteration of preference judgments (i.e., the annotator is presented with two sentences at a time, and is asked to chose one as more relevant to the task than the other).

In order to simplify the annotator's task, as well as construct a more reliable training set, we used the pairwise approach. Our decision was influenced by Carterette et al. (2008), who showed that preference judgments are faster and easier to make than absolute judgments. Thus, we can obtain many annotated instances in a relatively short amount of time. Moreover, since there are only two possible outcomes in choosing one sentence, we need at most three judges for a majority vote. This will also ensure consistency in the annotations. We discuss the model trained on preference judgments in Section 2.2.

2.2 Learned Models: $\textit{Pref}_{\text{SVM}}$ and $\textit{Rank}_{\text{Pref}}$

We used the preference judgments to learn a model, $\textit{Pref}_{\text{SVM}}$, that picks one sentence from a pair of sentences. This model was built using $\text{SVM}^{\textit{Light}}$ with a linear kernel. The examples used in the learning process correspond to pairs of sentences. For each pair, we constructed a vector of feature values, by subtracting the feature values corresponding to the first sentence from the feature values corresponding to the second sentence. We assigned a positive value to a pair vector if the first sentence was preferred and a negative value if the second one was preferred.

We can also use $\textit{Pref}_{\text{SVM}}$ to design a system that can rank all the sentences containing a gene and an iTerm, by performing comparisons between sentences in the list. We call $\textit{Rank}_{\text{Pref}}$ the system that picks one sentence from a group of sentences, and which also ranks the entire set of sentences. This method recursively applies $\textit{Pref}_{\text{SVM}}$ in the following manner: Two sentences are randomly picked from a given list of sentences. $\textit{Pref}_{\text{SVM}}$ chooses one sentence and discards the other. A third sentence is then randomly picked from the list, and $\textit{Pref}_{\text{SVM}}$ makes its choice by comparing it to the sentence kept in the previous step. This process of picking, comparing and discarding sentences is continued until there is only one sentence left. We keep track of comparison results and apply transitivity, in order to speed up the process of ranking all the sentences.

2.3 Features

Each sentence is first chunked into base phrases. We used Genia Tagger (Tsuruoka et al., 2005), which provides part-of-speech tags for every word in the sentence. We trained a chunker (i.e., shallow parser that identifies base NPs) using the Genia corpus.

We considered typical features that are used in machine learning approaches, such as distance between gene and iTerm, length of sentence, etc. Moreover, we included additional groups of features that we felt might be important for this task: one group to capture how the relationship is described textually, another group to capture how central the relationship is in terms of what is being described in the sentence, and the last to capture whether the relation is stated as a conjecture or a fact. The weights for these features will be determined automatically during the learning process and they will be dependent on whether or not the features were effective, given the annotation set.

The first type of features is to capture how the relationship is described textually. As an example, consider the sentence "*Bmp2* stimulates **osteoblastic differentiation**"[2], where the gene and the iTerm are in subject and object (direct object or otherwise) positions, and the verb is a common biological verb. Thus, we constructed a set of *lexico-syntactic patterns* to capture the different kinds of argument relations served by the two concepts. We grouped 25 lexico-syntactic patterns into 8 groups, corresponding to different relational constructions that can exist between a gene and an iTerm. Example patterns are shown in Table 1 for each group, and the symbols used in these patterns are explained in Table 2. When a sentence matches a pattern group, the corresponding value is set to 1 for that feature.

[2] In our examples, the gene will be marked in *italics* and the iTerm will be marked in **bold**.

Group	Example Pattern
G1	**G** VG+ **I**
G2	**G/I** via/by/through **I/G**
G3	**G** VG+ (NP/PP)* by/in VBG **I**
G4	**G/I** by/in VBG **I/G**
G5	**G/I** VB **I/G**
G6	**G/I** of **I/G**
G7	**G/I** other_preposition **I/G**
G8	including/such as/etc. **G/I** and **I/G**

Table 1: Examples of lexico-syntactic patterns

Symb	Definition
NP	a base noun phrase
PP	a preposition followed by a base noun phrase
VG+	a series of one or more verb groups
VBG	a verb group in which the head is a gerund verb
VBN	a verb group in which the head is a participle verb
VB	a verb group in which the head is a base verb
G, I	base noun phrases, with 0 or more prepositional phrases, containing the gene/iTerm

Table 2: Symbols used in the pattern notation

For example, the following sentence, in which the gene is *Lmo2* and the iTerm is "erythropoiesis", matches the pattern in G1: [**G** VG+ **I**].

> While Tal1 has been shown to induce erythroid differentiation , *Lmo2* appears to suppress fetal **erythropoiesis**.

where "Lmo2" matches **G**, "appears to suppress" matches VG+, and "fetal erythropoiesis" matches **I**.

Notice how the verb plays an important role in the patterns of groups G1, G3, G4, and G5. We also have a *verb type* feature which differentiates groups of verbs having the gene and the iTerm as arguments (e.g., "activates", "is involved in", "plays a role", etc. are treated as different types).

The second type of features captures how central the relationship is in terms of what is being described in the sentence. The *subject feature* records whether the gene and iTerm appear in the subject position, as this will tell us if they are in focus in the sentence. While we do not parse the sentence, we take a simplified sentence (see Section 2.4) and see if the gene/term appear in a noun phrase preceding the first tensed verb. Another feature, the *gene-iTerm position*, measures how close the gene and the term are to each other and to the beginning of the sentence, as this makes it easier for a reader to grasp the relation between them. For this, we add the number of words occurring to the left of the segment spanning the gene and iTerm, and half of the number of words occurring between them. Finally, we included a *headedness feature*. The idea here is that if the gene/term are not the head of the noun group, but rather embedded inside, then this potentially makes the relation less straightforward. These groups are denoted by **G** and **I** in the patterns shown in Table 1.

The third type of features captures information about the sentence itself. The *sentence complexity* feature is measured in terms of the number of verbs, conjunctions, commas, and parentheticals that occur in the sentence. We use a *conjecture* feature for detecting whether the sentence involves a hypothesis. We have a simple rule for this feature, by seeing if words such as "may", "could", "probably", "potentially", etc., appear in proximity of the gene and iTerm. Additionally, we have a *negation* feature to detect whether the relationship is mentioned in a negative way. We look for words such as "not", "neither", etc., within proximity of the gene and iTerm.

Although the features and lexico-syntactic patterns were determined by analyzing a development set of sentences containing genes and their iTerms, we believe that these features and patterns can be used to rank sentences involving other biomedical entities, not just genes.

2.4 Sentence Simplification

Notice that the lexico-syntactic patterns are written as sequences of chunks and lexical tags. If a sentence matches a pattern, then the sentence expresses a relation between the gene and the iTerm. However, sometimes it is not possible to match a pattern if the sentence is complex.

For example, consider sentence A in Table 3, for gene *Cd63*. Let us assume that the iTerm is "protasomes". Clearly, there is a relationship between the gene and the iTerm, namely that *Cd63* was found in pc-3 cell-derived protasomes. However, none of the lexico-syntactic patterns is able to capture this relation, because of all the extra information between

A	*Cd63*, an integral membrane protein found in multivesicular lysosomes and **secretory granules**, was also found in pc-3 cell-derived **protasomes**.
S1	*Cd63* was found in pc-3 cell-derived **protasomes**.
S2	*Cd63* is an integral membrane protein.
CS1	*Cd63* is found in multivesicular lysosomes.
CS2	*Cd63* is found in **secretory granules**.

Table 3: Simplified sentences for gene *Cd63*. Example iTerms: “protasomes” and “secretory granules”.

the gene and the term. While we may have multiple patterns in each group, we cannot necessarily account for each lexical variation at this level of granularity.

We are using a sentence simplifier, built in-house, to ensure a match where applicable. The simplifier identifies appositions, relative clauses, and conjunctions/lists of different types, using regular expressions to match chunked tags. In the sentence of Table 3, the simplifier recognizes the apposition “an integral membrane protein”, the reduced relative clause “found in multivesicular bodies/lysosomes and secretory granules” and the noun conjunction “multivesicular bodies/lysosome and secretory granules”. It then produces several simplified sentences containing the gene. S1 and S2, shown in Table 3, are simplified sentences obtained from the simplifier. CS1 and CS2 are additional simplified sentences, which required the combination of multiple simplifications: the appositive, the relative clause, and the noun conjunction.

Notice how each of the simplified sentences shown in Table 3 is now matching a pattern group. If we are interested in the relationship between *Cd63* and “protasomes”, we can look at S1. Likewise, if we are interested in the relationship between *Cd63* and “secretory granules”, we can look at CS2.

We have a *matching* feature that tells whether the pattern was matched in the original sentence, a simplified sentence, or a combined sentence, and this feature is taken into account in the learning process.

3 Results and Discussion

We evaluated both $Pref_{\text{SVM}}$ and $Rank_{\text{Pref}}$. Each required a different set of annotated data. For the evaluation of $Pref_{\text{SVM}}$, we used the preference judgments and leave-one-out cross validation. And for the evaluation of $Rank_{\text{Pref}}$, we asked the annotators to order a group of sentences mentioning gene-iTerm pairs. Six life science researchers, with graduate degrees, annotated both sets.

3.1 Evaluation of $Pref_{\text{SVM}}$

First, we evaluated the performance of $Pref_{\text{SVM}}$ using leave-one-out cross validation.

3.1.1 Annotation of Preference Judgements

We started by selecting a group of pairs of sentences. We randomly picked gene-iTerm combinations, and for each combination, we randomly picked two sentences containing both the gene and the term. To alleviate bias, the order of the sentences was chosen randomly before displaying them to the annotators. In our guidelines, we asked the annotators to choose sentences that clearly state the relationship between the gene and the iTerm. Because the focus here is on the relationship between the two terms, we also asked them to refrain from choosing sentences that describe additional information or other aspects. It is conceivable that, for other applications, extra information might be an important determining factor, but for our task we wanted to focus on the relationship only.

For each pair of sentences, we wanted to have three opinions so that we can have a majority vote. To alleviate the burden on the annotators, we started by giving each pair of sentences to two annotators, and asked for an extra opinion only when they did not agree. Each biologist was given an initial set of 75 pairs of sentences to annotate, and shared the same amount of annotations (15) with each of the other biologists. 225 unique pairs of sentences were thus annotated, but six were discarded after the annotators informed us that they did not contain the gene in question.

In 34 out of 219 pairs of sentences, the two biologists disagreed on their annotations. These cases included pairs of similar sentences, or pairs of sentences that did not describe any relationship between

System	Performance	Correct
Baseline 1	65.75%	144
Baseline 2	71.69%	157
Pref$_{\text{SVM}}$ without Simp	72.14%	158
Pref$_{\text{SVM}}$ with Simp	83.10%	182

Table 4: Results for *Pref*$_{\text{SVM}}$

the gene and the iTerm. An example of sentences for which the annotators could not agree is:

> 1. The tle proteins are the mammalian homologues of *gro*, a member of the drosophila **notch signaling** pathway.
> 2. In drosophila, *gro* is one of the neurogenic genes that participates in the **notch signalling** pathway .

For these 34 pairs, we randomly selected another annotator and considered the majority vote.

3.1.2 Baselines

We chose two baselines against which to compare *Pref*$_{\text{SVM}}$. The first baseline always chooses the shortest sentence. For the second baseline, we looked at the proximity of the gene/term to the beginning of the sentence, as well as the proximity of the two to each other, and chose the sentence that had the lowest accumulated proximity. The reason for this second baseline is because the proximity of the gene/term to the beginning of the sentence could mean that the sentence focuses on the gene/term and their relation. Furthermore, the proximity of the gene to the iTerm could mean a clearer relation between them.

3.1.3 Results

We evaluated *Pref*$_{\text{SVM}}$ by performing leave-one-out cross validation on the set of 219 pairs of sentences. Each pair of sentences was tested by using the model trained on the remaining 218 pairs. The results are shown in Table 4.

The first baseline performed at 65.75%, correctly choosing 144 of 219 sentences. The second baseline performed slightly better, at 71.69%. *Pref*$_{\text{SVM}}$ outperformed both baselines, especially when the sentence simplifier was used, as this facilitated the match of the lexico-syntactic patterns used as features. *Pref*$_{\text{SVM}}$ performed at 83.10%, which is 17.35% better than the first baseline, and 11.41% better than the second baseline.

3.2 Evaluation of *Rank*$_{\text{Pref}}$

The previous evaluation showed how *Pref*$_{\text{SVM}}$ performs at picking a sentence from a pair of sentences. But ultimately, for the intended eGIFT application, the system needs to choose one sentence from many. We evaluated *Rank*$_{\text{Pref}}$ for this task.

3.2.1 Annotating Data for Sentence Selection

For this evaluation, we needed to create a different set of annotated data that reflects the selection of one sentence from a group of sentences.

Since a gene and an iTerm can appear in many sentences, it is too onerous a task for a human annotator to choose one out of tens or hundreds of sentences. For this reason, we limited the set of sentences mentioning a gene and an iTerm to only 10. We randomly picked 100 gene-term pairs and for the pairs that contained more than ten sentences, we randomly chose ten of them. On average, there were 9.4 sentences per set.

We asked the same annotators as in the previous evaluation to participate in this annotation task. Because the task is very time consuming, and because it is hard to decide how to combine the results from multiple annotators, we assigned each set of sentences to only one annotator. We showed the sentences in a random order so that biasing them would not be an issue.

We initially asked the annotators to order the sentences in the set. However, this task proved to be impossible, since many sentences were alike. Instead, we asked the annotators to assign them one of three categories:

(Cat.1) Any sentence in this category could be considered the "best" among the choices provided;

(Cat.2) These sentences are good, but there are other sentences that are slightly better;

(Cat.3) These sentences are not good or at least there are other sentences in this set that are much better.

Classifying the sentences into these categories was less cumbersome, fact which was confirmed by our evaluators after a trial annotation.

Out of the total of 936 sentences, 322 (34.4%) were placed in the first category, 332 (35.5%) were

System	Cat.1	Cat.2	Cat.3
Baseline 1	58	30	12
Baseline 2	61	24	15
Rank$_{\mathrm{Pref}}$ without Simp	67	21	12
Rank$_{\mathrm{Pref}}$ with Simp	80	17	3

Table 5: Results for *Rank*$_{\mathrm{Pref}}$

placed in the second category, and 282 (30.1%) were placed in the third category. On average, it took about 15 minutes for an annotator to group a set's sentences into these three categories. So each annotator volunteered approximately 5 hours of annotation time.

3.2.2 Results

Table 5 shows how the top sentences picked for the 100 gene-term pairs by the four systems matched with the annotations. 80 of 100 sentences that *Rank*$_{\mathrm{Pref}}$ picked were placed in Cat.1 by the annotators, 17 were placed in Cat.2, and 3 sentences were placed in Cat.3. These results compare favorably with results obtained for the two baselines and *Rank*$_{\mathrm{Pref}}$ without the use of the simplifier.

Furthermore, instead of just focussing on the top choice sentence, we also considered the ranking of the entire set of sentences. We looked at how the ranked lists agree with the categories assigned by the annotators. We used the normalized discounted cumulative gain (nDCG) (Jarvelin and Kekalainen, 2002), a standard metric used in information retrieval to evaluate the quality of the ranked lists. DCG at rank p is defined as:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2 i}$$

where rel_i is the relevance of the item at position i. We normalize DCG by dividing it by an ideal gain (i.e., DCG of same list, when ordered from highest to lowest relevance).

For our task, we took the relevance score to be 1 for a sentence placed in Cat.1, a relevance score of 0.5 for a sentence placed in Cat.2, and a relevance score of 0 for a sentence placed in Cat.3. We report a normalized discounted cumulative gain of 77.19%.

This result compares favorably with results reported for the two baselines (68.36% for B1 and 68.32% for B2) as well as for when the sentence simplifier was removed (69.45%).

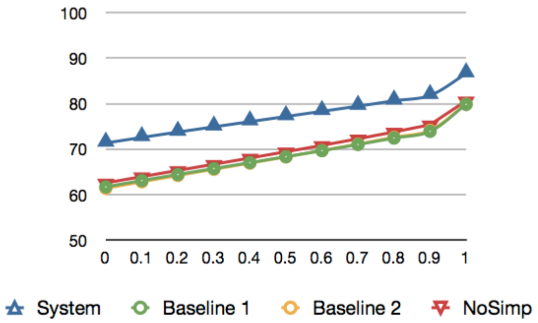


Figure 2: Distribution of nDCG for different relevance scores assigned to sentences placed in category Cat.2.

Figure 2 shows different results for nDCG when the relevance score for Cat.2 is varied between 0 (same as sentences placed in Cat.1) and 1 (same as sentences placed in Cat.3).

4 Related Work

To the best of our knowledge, no one has attempted to rank sentences from the biomedical literature, using machine learning on a set of data marked with preference judgments. However, different approaches have been described in the literature that use preference judgments to learn ranked lists. For example, Radlinski and Joachims (2005) used preference judgments to learn ranked retrieval functions for web search results. These judgments were generated automatically from search engine logs. Their learned rankings outperformed a static ranking function. Similar approaches in IR are those of Cohen et al. (1999) and Freund et al. (2003).

Ranking of text passages and documents has been done previously in BioNLP for other purposes. Suomela and Andrade (2005) proposed a way to rank the entire PubMed database, given a large training set for a specific topic. Goldberg et al. (2008) and Lu et al. (2009) describe in detail how they identified and ranked passages for the 2006 Trec Genomics Track (Hersh et al., 2006). Yeganova et al. (2011) present a method for ranking positively labeled data within large sets of data, and this method was applied by Neveol et al. (2011) to rank sentences containing deposition relationships between biological data and public repositories.

Extraction of sentences describing gene functions has also been applied for creating gene summaries (Ling et al., 2007; Jin et al., 2009; Yang et al., 2009). However, these methods differ in that their goal is not to look for sentences containing specific terms and their relations with genes, but rather for sentences that fall into some predefined categories of sentences typically observed in gene summaries.

Sentence simplification has been used to aid parsing (Chandrasekar et al., 1996; Jonnalagadda et al., 2009). Devlin and Tait (1998) and Carroll et al. (1998) use it to help people with aphasia. Siddharthan (2004) was concerned with cohesion and suggested some applications.

The idea of using lexico-syntactic patterns to identify relation candidates has also been applied in the work of Banko et al. (2007), although their patterns are not used in the learning process.

5 Conclusion and Future Directions

We have developed a system which aims to identify sentences that clearly and succinctly describe the relation between two entities. We used a set of preference judgements, as provided by biologists, to learn an SVM model that could make a choice between any two sentences mentioning these entities.

The model compares favorably with baselines on both the task of choosing between two sentences, as well as ranking a set of sentences. The performance for choosing between two sentences was 83.10%, as compared to 65.75% and 71.69% for the two baselines, respectively. For choosing one sentence from a list of sentences, the performance was 80%, as compared to 58% and 61%. Furthermore, when the entire list of ranked sentences was evaluated, the system reported a nDCG of 77.19%, compared to 68.36% and 68.32% for the two baselines.

The model's performance was also shown to be significantly better when sentence simplification was used. We were able to match relation patterns on complex sentences, and observed an increase of 10.96%, 13%, and 7.74% for the three evaluations afore-mentioned, respectively. It is noteworthy that, without the simplification, the performance is only slightly better than the second baseline. This is because the second baseline uses information that is also used by our system, although this does not include the lexico-syntactic patterns that identify the type of relation between the gene and the term.

Given that the full system's performance is much better than both baselines, and that the system's performance without simplification is only slightly better than the second baseline, we believe that: (1) the pattern and type of relation determination are important, and (2) sentence simplification is crucial for the determination of the relationship type.

We are currently pursuing summaries for genes. Since iTerms have been shown in previous evaluations to represent important aspects of a gene's functionality and behavior, we are investigating whether they are represented in gene summaries found in EntrezGene and UniProtKB. If so, an *extractive* summary can be produced by choosing sentences for the gene and its iTerms. We are also considering developing *abstractive* summaries. Our use of lexico-syntactic patterns can be extended to pick the exact relation between a gene and the iTerm. For example, by using the lexico-syntactic patterns, coupled with simplification, we can extract the following exact relations from the four sentences shown in Figure 1: "Groucho is a corepressor", "The wrpw motif recruits groucho", "Groucho is implicated in notch signaling", and "The eh1 repression domain binds groucho". With these relations extracted, using text generation algorithms for textual realization and cohesion, we can produce *abstractive* summaries.

We would also like to investigate how to generalize this work to other pairs of entities, as well as how to generalize this work for other applications which may or may not require the same features as the ones we used.

Acknowledgments

This work has been supported in part by USDA Grant 2008-35205-18734 and the Agriculture and Food Research Initiative Competitive USDA Grant 2011-67015-3032. We thank Cecilia Arighi, Kevin Bullaughey, Teresia Buza, Fiona McCarthy, Lakshmi Pillai, Carl Schmidt, Liang Sun, Hui Wang, and Qinghua Wang for participating in the annotation task and/or for various discussions. We also thank the anonymous reviewers for their comments and suggestions, which helped us improve the manuscript.

References

Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open Information Extraction from the Web. In *Proceedings of IJCAI*.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of English newspaper text to assist aphasic readers. *Proceedings of the AAAI98 Workshop on Integrating AI and Assistive Technology*, pages 7–10.

Ben Carterette, Paul N Bennett, David Maxwell Chickering, and Susan T Dumais. 2008. Here or there: Preference judgments for relevance. In *Proceedings of the IR research, 30th European conference on Adv. in IR*.

R Chandrasekar, Christine Doran, and B Srinivas. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th conference on Computational linguistics*, volume 2, pages 1041–1044. Association for Computational Linguistics.

Wiliam W Cohen, Robert E Schapire, and Yoram Singer. 1999. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270.

Siobhan Devlin and John Tait. 1998. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic Databases*, pages 161–173.

Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969.

Arek Gladki, Pawel Siedlecki, Szymon Kaczanowski, and Piotr Zielenkewicz. 2008. e-LiSe–an online tool for finding needles in the 'Medline haystack'. *Bioinformatics*, 24(8):1115–1117.

Andrew B Goldberg, David Andrzejewski, Jurgen Van Gael, Burr Settles, Xiaojin Zhu, and Mark Craven. 2008. Ranking biomedical passages for relevance and diversity. In *Proceedings of TREC*.

William Hersh, Aaron M Cohen, Phoebe Roberts, and Hari Krishna Rekapalli. 2006. TREC 2006 Genomics Track Overview.

Kalervo Jarvelin and Jaana Kekalainen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446.

Feng Jin, Minlie Huang, Zhiyong Lu, and Xiaoyan Zhu. 2009. Towards automatic generation of gene summary. In *Proceedings of the BioNLP 2009 Workshop*, pages 97–105. Association for Computational Linguistics, June.

Siddhartha Jonnalagadda, Luis Tari, Jorg Hakenberg, Chitta Baral, and Graciela Gonzalez. 2009. Towards effective sentence simplification for automatic processing of biomedical text. In *Proceedings of NAACL HLT 2009: Short Papers*, pages 177–180.

Szymon Kaczanowski, Pawel Siedlecki, and Piotr Zielenkewicz. 2009. The high throughput sequence annotation service (HT-SAS) - the shortcut from sequence to true medline words. *BMC Bioinformatics*, 10:148–154, May.

Jung-Jae Kim, Piotr Pezik, and Dietrich Rebholz-Schuhmann. 2008. MedEvi: Retrieving textual evidence of relations between biomedical concepts from medline. *Bioinformatics*, 24(11):1410–1412.

Xu Ling, Jing Jiang, Xin He, Qiaozhu Mei, Chengxiang Zhai, and Bruce Schatz. 2007. Generating gene summaries from biomedical literature: A study of semi-structured summarization. *Information Processing and Management*, 43:1777–1791, March.

Yue Lu, Hui Fang, and Chengxiang Zhai. 2009. An empirical study of gene synonym query expansion in biomedical information retrieval. *Information Retrieval*, 12:51–68, February.

Aurélie Névéol, W John Wilbur, and Zhiyong Lu. 2011. Extraction of data deposition statements from the literature: a method for automatically tracking research results. *Bioinformatics*, 27(23):3306–3312.

Filip Radlinski and Thorsten Joachims. 2005. Query chains: Learning to rank from implicit feedback. In *Proceedings of KDD'05*.

Advaith Siddharthan. 2004. *Syntactic Simplification and Text Cohesion*. Ph.D. thesis, University of Cambridge.

Neil R Smalheiser, Wei Zhou, and Vetle I Torvik. 2008. Anne O'Tate: A tool to support user-driven summarization, drill-down and browsing of PubMed search results. *Journal of Biomedical Discovery and Collaboration*, 3(1):2–11.

Brian P Suomela and Miguel A Andrade. 2005. Ranking the whole MEDLINE database according to a large training set using text indexing. *BMC Bioinformatics*, 6(75), March.

Yoshimasa Tsuruoka, Yuka Tateishi, Jing-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. In *Advances in Informatics – 10th Panhellenic Conference on Informatics, LNCS 3746*, pages 382–392.

Catalina O Tudor, Carl J Schmidt, and K Vijay-Shanker. 2010. eGIFT: Mining Gene Information from the Literature. *BMC Bioinformatics*, 11:418.

Jianji Yang, Aaron Cohen, and William Hersh. 2009. Evaluation of a gene information summarization system by users during the analysis process of microarray datasets. *BMC Bioinformatics*, 10(Suppl 2):S5.

Lana Yeganova, Donald C Comeau, Won Kim, and W John Wilbur. 2011. Text Mining Techniques for Leveraging Positively Labeled Data. In *Proceedings of ACL Workshop BioNLP*, pages 155–163.