RC24060 (W0609-113) September 25, 2006
Computer Science

IBM Research Report

Computing Similarities between Natural Language Descriptions of Knowledge and Skills

Robert G. Farrell
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Feng Pan
Information Sciences Institute
University of Southern California

Computing Similarities between Natural Language Descriptions of Knowledge and Skills

Robert G. Farrell
IBM T J Watson Research Center
robfarr@us.ibm.com

Feng Pan
Information Sciences Institute, University of Southern California
pan@isi.edu

Abstract: This paper explores the problem of computing text similarity utilizing natural language processing. Four parsers are evaluated on a large corpus of skill statements from a corporate expertise taxonomy. A similarity measure utilizing common semantic role features extracted from parse trees was found superior to an information-theoretic measure of similarity and comparable to human judgments of similarity.

1. Introduction

Knowledge-intensive industries need to become more efficient at deploying the right expertise as quickly and efficiently as possible. At IBM, the Professional Marketplace system offers a single view of employees in IBM Global Services in order to quickly match and deploy skilled individuals to meet customer needs. Since the introduction of this new job search system, engagements have been staffed 20% faster. Anecdotal data suggests that deployed individuals are also better matched to exact qualifications requested by the client. In addition, there is nearly a 10% decrease in the use of subcontractors due to better utilization of the IBM workforce. Improved efficiencies in workforce management have saved the company over US$500 million.[1]

The IBM Professional Marketplace depends upon a centralized database of employee profiles, including job roles and skill sets. While in some industries the skills for each job role can be enumerated, at IBM and other knowledge-intensive companies the process of tracking employee skills is more difficult. First, employees typically take on many different assignments and develop a broad range of skills across multiple job roles. Second, although there is a common skills dictionary, it is still hard for employees to find the perfect skills to describe their skill sets. For example, an employee might not know whether to choose a skill stating that they "maintain" a given product or "support" it or whether to choose a skill about maintaining a "database" or about maintaining "DB2".

The IBM Professional Marketplace offers a powerful search feature to find employees to match open positions. Users can search on location, availability dates, job roles, skills, and other criteria. However, the searches are only based on exact matches to the skills

[1] Professional Marketplace. Published on 07/24/2006. http://www-306.ibm.com/software/success/cssdb.nsf/CS/LJKS 6RMJZS?OpenDocument&Site=

dictionary. Exact matching is very likely to miss employees who are very good matches to the position but didn't select the exact skills that appeared in the open job position. Managers searching for employees to fill positions may miss qualified applicants.

Thus, it is desirable for the job search system to be able to find *approximate matches*, instead of only exact matches, between available employees and open job positions. More specifically, a skill affinity computation is needed to allow searches to be expanded to related skills, and return more potential matches.

In this paper, we present our preliminary work on developing a skill affinity computation based upon semantic similarities between skills. In this work, a skill is the ability to perform an action, such as advising or leading, to some level of proficiency. We demonstrate that we can improve on standard statistical text similarity techniques by utilizing natural language processing. In Section 2, we first describe IBM's expertise taxonomy which provides a hierarchical organization of over 10,000 skills. We then describe in Section 3 how we identify and assign semantic roles for skill descriptions, and match skills on corresponding roles. We compared and evaluated four natural language parsers (the IBM ESG parser, the Charniak parser, the Stanford parser, and MINIPAR) for the purpose of our task. The semantic similarity computation between skill verbs will also be described. The inter-rater agreement study and the evaluation results of our approach will be presented in Section 4.

2. The IBM Expertise Taxonomy

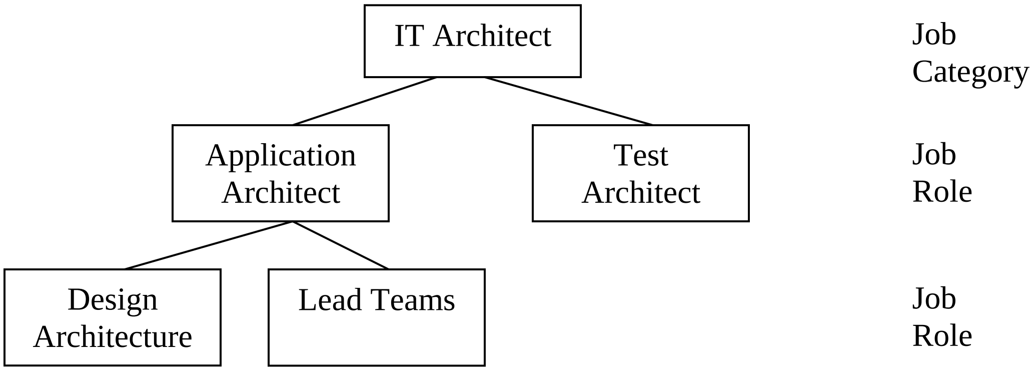

Figure 1. IBM Expertise Taxonomy Example

IBM expertise taxonomy is a standardized, enterprise-wide, language and structure to describe job role requirements and people capabilities (skill sets) across IBM. In the taxonomy, skills are associated with job roles. For example, Figure 1 shows a portion of the taxonomy where a set of skills are displayed under the job role of "Application Architect" which is associated with the job category "IT Architect".

The IBM expertise taxonomy contained 10667 skills when we started our work. Each skill contains a title and a more detailed description.[2] For example, here is the skill "Advise BAAN eBusiness ASP"

> *Title*: Advise BAAN eBusiness ASP
> *Description*: -Advise and counsel the customer regarding the product/situation/ solution. –Understand and recommend actions that may be considered to resolve the customer's problems or issues within this area. -Direct client experience with the product

Taxonomy update policies require that skill titles be verb phrases using one of 20 valid skill verbs, including *Advise, Architect, Code, Design, Implement, Release, Sell, and Support.* However, a total of 82 verbs are actually used in the 10667 skill titles retrieved from the taxonomy.

The distribution analysis of the skill verbs shows that the 19 valid verbs ("Release" is not used in the taxonomy) covers 10562 skills which accommodate more than *99%* of the total skills. Thus we concentrated our effort on the 19 valid verbs One interesting observation is that though "Architect" is treated as a verb in the taxonomy, it has no verb sense in WordNet (Miller, 1990) and many other dictionaries[3].

3. Computing Semantic Similarities between Skill Descriptions

There are several techniques that can be used to compute semantic similarities between skills. In this work we examined both statistical techniques and natural language processing. In the next section, we explain our statistical approach based on information theory. In the remainder, we describe how we use natural language processing techniques to extract semantic role information from the skill descriptions. In this work, a semantic role is the underlying relationship between the objects, participants, or concepts mentioned in a skill description and the skill verb indicating what action is to be performed,

3.1 Statistical Approach

In order to compute semantic similarities between skill descriptions, we first adopted one of the standard statistical approaches to the problem of computing text similarities based on Lin's information-theoretic similarity measure (Lin 1998a) which is a universal similarity measure that doesn't presume any form of knowledge representation.

Lin defined the commonality between A and B as

$$I(common(A,B))$$

[2] Some of the skill descriptions are the same as their titles.
[3] "Architect" does have a verb sense in the Oxford English Dictionary (http://dictionary.oed.com/).

where *common*(*A, B*) is a proportion that states the commonalities between A and B and where *I*(*s*) is the amount of information contained in a proposition *s* which can be measured by the negative logarithm of the probability of the proposition *s*.

The similarity between A and B is then defined as the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are:

$$Sim(A,B) = \frac{\log P(common(A,B))}{\log P(description(A,B))}$$

In order to compute *common*(*A, B*) and *description*(*A,B*), we use a bag of words as features, i.e., the unigram features -- the frequency of words computed from the entire corpus of the skill descriptions. Thus *common*(*A,B*) is the unigrams that both skill descriptions share, and *description*(*A,B*) is the union of the unigrams from both skill descriptions.

The words are stemmed first so that the words with the same root (e.g., managing & management) can be found as commonalities between two skill descriptions. A stop-word list is also used so that the commonly used words in most of the documents (e.g., the, a) are not used as features.

A more formal evaluation of this approach will be presented in Section 4 where the similarity results for 75 pairs of skills will be evaluated against human judgments. In order to see how to improve this standard statistical approach, we examined sample skill pairs which achieved high similarity scores from the statistical approach but don't seem so similar to humans in our evaluation[4]:

(1) Advise Business Knowledge of *CAD functionality* for FEM
Advise on Business Knowledge of *Process* for FEM

(2) Advise on *Money Market*
Advise on *Money Center Banking*

In both examples, although many words are shared between the two pairs of skills (i.e., "Advise Business Knowledge of ... for FEM" in (1); "Advise on Money" in (2)), they are not so similar since the key components of the skills (i.e., "CAD functionality" vs. "Process" in (1); "Money Market" vs. "Money Center Banking" in (2)) are different.

Thus, we can see that the similarity computation would be more accurate if it matches on corresponding semantic roles, instead of matching key words from any places in the skill descriptions. We also want to concentrate more on the matching between important semantic roles, i.e., the key components of the skills.

3.2 Identifying and Assigning Semantic Roles

[4] Here we only show the titles of the skill descriptions.

The following example shows the kind of semantic roles we want to be able to identify and assign.

> [$_{\textbf{action}}$ Apply] [$_{\textbf{theme}}$ Knowledge of [$_{\textbf{concept}}$ IBM E-business Middleware]] to
> [$_{\textbf{purpose}}$ PLM Solutions]

In this example, “Apply” is the “action” of the skill; “Knowledge of IBM E-business Middleware” is the “theme” of the skill, where the “concept” semantic role (i.e., “IBM E-business Middleware”) specifies the key component of the skill requirements and is the most important role for the skill matching; “PLM Solutions” is the “purpose” of the skill.

Our goal is to extract all such semantic role patterns for all the skill descriptions. We started with the skill titles first, which summarize the skill descriptions and use a limited number of skill verbs.

Although there exists some automatic semantic role taggers (Gildea & Jurafsky, 2002; Xue & Palmer, 2004; Pradhan et. al., 2004, 2005; Giuglea & Moschitti, 2006), most of them were trained on PropBank (Palmer et. al., 2005) and/or FrameNet (Johnson et. al., 2003), and perform much worse in different corpora (Pradhan et. al., 2004). Our corpus is from such a different domain and there are many domain-specific terms in the skill descriptions. Given this, we would expect an even worse performance from these automatic semantic role taggers. Moreover, the semantic role information we need to extract is more detailed and deep than most of the automatic semantic role taggers can identify and extract (e.g., the “concept” role is embedded in the “theme” role).

For our task, since the 19 valid skill verbs cover more than 99% of all the 10667 skills, we can afford to develop a domain-specific semantic role parser which can extract semantic role patterns from each of those 19 skill verbs, which will definitely achieve a much higher performance. The input needed for the semantic role parser is syntactic parse trees generated by a syntactic parser from the original skill titles in natural language.

3.3 Preprocessing for Parsing

We first used the Charniak parser (2000) to parse the original skill titles. However, among all the 10667 titles, 1217 of them were not parsed as verb phrases, an a priori requirement, After examining the error cases, we found that abbreviations are used very widely in the skill titles. For example, “Advise Solns Supp Bus Proc Reeng for E&E Eng Procs”. So the first step of the preprocessing was to expand abbreviations.

There are 225 valid abbreviations identified by the expertise taxonomy team. However, we found many abbreviations that appeared in the skill titles but were not listed there. Since most of the abbreviations are not words in a dictionary, in order to find the abbreviations that appear frequently in the skill titles, we first found all the words in the skill titles that were not in WordNet. We then ranked them based on their frequencies, and manually found the possible high frequency abbreviations. By this approach, we added another 187 abbreviations to the list (a total of 412).

We also found from the error cases that many words were mistagged as proper nouns, For example, “Technically” in “Advise Technically for Simulation” was parsed as a proper noun. We realized the reason for this error was that all the words, except for prepositions, are capitalized in the original titles, and the parser tends to tag them as proper nouns. To solve this problem, we changed all the capitalized words to lower case, except for the first word and the acronyms (words that have all letters capitalized (e.g., IBM)).

After applying these two steps of preprocessing (i.e., abbreviation expansion & converting capitalized words to lower case), we parse the skill titles again. This time, 1007 skills were not parsed as verb phrases -- more than 200 additional skills were parsed as verb phrases after the preprocessing. This was quite promising.

When we examined the error cases more closely this time, we found that the errors occur mostly when the skill verbs can be both a noun and a verb (e.g., design, plan). In those cases, the parser may parse the entire title as one noun phrase, instead of a verb phrase. In order to disambiguate such cases, we added a subject (“Employees”) to all the skill titles to convert all verb phrases into full sentences.

After applying this additional step of preprocessing, we parsed the skill titles again. This time, only 28 skills were not parsed as verb phrases, which is a significant improvement. Those remaining errors were due to the word “Architect” which, as mentioned in Section 2, has no verb sense in WordNet.

3.4 Parser Evaluation and Comparison

While Charniak’s parser performed well in our initial verb phrase (VP) test, we decided to test its syntactic parsing performance in more detail. More importantly, we decided to compare the Charniak parser’s performance to other popular parsers . For this evaluation, we also compared the Stanford parser, the IBM ESG parser, and MINIPAR.

Stanford parser (Klein & Manning, 2003) is an unlexicalized statistical syntactic parser that was also trained on the Penn TreeBank. Its parse tree has the same structure as the Charniak parser. For example, the parse output of the sentence “Employees apply business intelligence to finance.” is as follows, and the parse tree structure is shown in Figure 2.

```
(S1 (S (NP (NNS Employees))
    (VP (VBP apply)
       (NP (NN business) (NN intelligence))
       (PP (TO to) (NP (NN finance)))) (. .)))
```

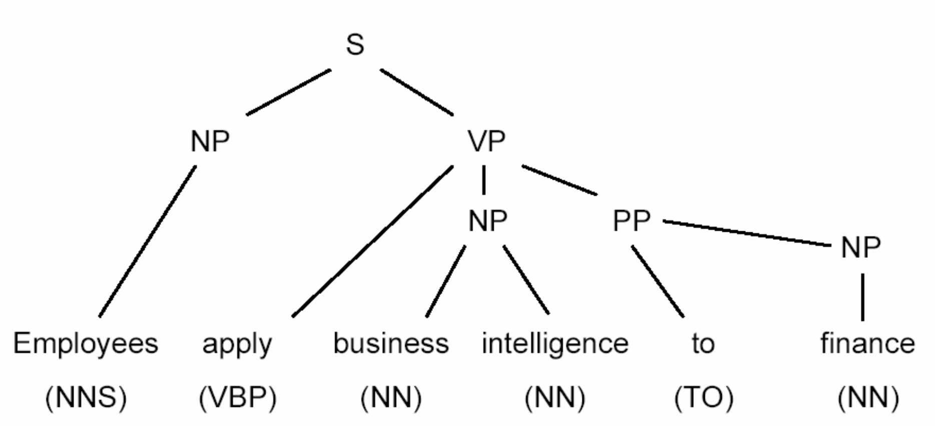

Figure 2. The Parse Tree of “Employees apply business intelligence to finance.”

IBM ESG (English Slot Grammar) **parser** (McCord, 1989) is a rule-based parser based on the slot grammar where each phrase has a head and dependent elements, and is also marked with a syntactic role. The slot filling for the sentence “Mary gave a book to John” is shown in Figure 3, from which we can see, for example, “gave” is the head of the sentence and it has a subject of “Mary”.

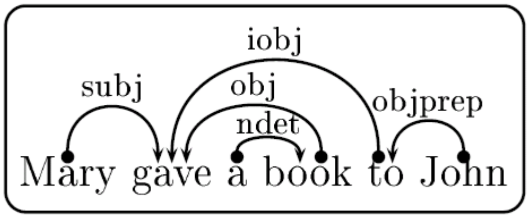

Figure 3. Slot Filling for “Mary gave a book to John.”

MINIPAR (Lin, 1998b), as a dependency parser, is very similar to the IBM ESG parser in terms of its output. It represents sentence structures as a set of dependency relationships. For example, the parse output of the sentence “Employees apply business intelligence to finance.” is as follows, and the dependency tree structure is shown in Figure 4.

```
( E0 (() fin C * )
1 (Employees employee N 2 s (gov apply))
2 (apply ~ V E0 i (gov fin))
E2 (() employee N 2 subj (gov apply) (antecedent 1))
3 (business ~ A 4 mod (gov intelligence))
4 (intelligence ~ N 2 obj (gov apply))
5 (to ~ Prep 2 mod (gov apply))
6 (finance ~ N 5 pcomp-n (gov to))
7 (. ~ U * punc) )
```

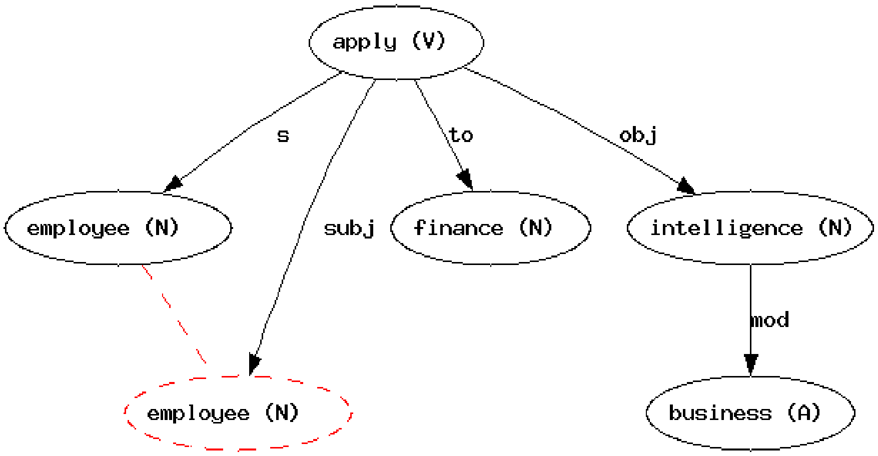

Figure 4. The Dependency Tree of “Employees apply business intelligence to finance.”

Since our purpose is to use the syntactic parses as inputs to extract semantic role patterns, the correctness of the bracketing of the parses and the POS (Part Of Speech) of the phrases (NP or VP) are the most important information for us, whereas the POS of individual words (e.g., nouns vs. proper nouns) is not that important (also, there are too many domain-specific terms in our data).

Thus, our evaluation of the parses is only on the correctness of the bracketing and the POS of the phrases (NP or VP), not the total correctness of the parses. To our task, the correctness of the prepositional attachments is especially important for extracting accurate semantic role patterns. For example, for the sentence

Apply Knowledge of IBM E-business Middleware to PLM Solutions.

the correct bracketing should be

Apply [Knowledge [of [IBM E-business Middleware]]] [to [PLM Solutions]].

thus the parser needs to be able to correctly attach “of IBM E-business Middleware” to “Knowledge” and attach “to PLM Solutions” to “Apply”, not “Knowledge”.

To evaluate the performance of the parsers, we randomly picked 100 skill titles from our corpus, preprocessed them, and then parsed them using the four different parsers. We then evaluated the parses using the above evaluation measures. The parses were rated as correct or not. No partial score was given.

Table 1 shows the evaluation results, where both the published accuracy and the accuracy for our task are listed.

	Published Accuracy	Accuracy for Our Task

IBM XSG	N/A	72%
Charniak's	90.1%	76%
Stanford's	86.9%	68%
MINIPAR	88.5%	49%

Table 1. Parser Evaluation and Comparison Results.

From the results, we can see that all the parsers perform worse for our task than their published results. After analyzing the error cases, we found out the reasons are

(1) Many domain specific terms and acronyms. For example, "SAP" in "Employees advise on *SAP* R/3 logistics basic data." was always tagged as verb by the parsers.
(2) Many long noun phrases. For example, "Employees perform *JD edwards foundation suite address book*."
(3) Some specialized use of punctuation. For example, "Employees perform business transportation consultant-logistics.sys."
(4) Prepositional attachments can be difficult. For example, in "Employees apply IBM infrastructure knowledge *for IDBS*", "for IDBS" should attach to "apply", but many parsers mistakenly attach it to "IBM infrastructure knowledge".

Compared with other parsers, we noticed that MINIPAR performs much worse than its published results. The main reason is that it always parses the phrase "VERB knowledge of Y" (e.g., "Employees *apply knowledge of web technologies*.") incorrectly, i.e., the parse result always mistakenly attaches "Y" (e.g., "web technologies") to the VERB (e.g., "apply"), not "knowledge". Since there were so many such phrases in the test set and in the corpus, this kind of error significantly reduced the performance for our task.

From the evaluation and comparison results we can see that the Charniak parser performs the best for our domain. Although the IBM ESG parser performs a little bit worse than the Charniak parser, its parses contain much richer syntactic (e.g., subject, object) and semantic (e.g., word sense) slot filling information, which can be very useful to many natural language applications. Since our goal is to use the syntactic parses to extract semantic role patterns, the bracketing information (i.e., what we evaluated on) is the most important factor. Thus, we decided to use the Charniak parser for our task.

3.5 Extracted Semantic Role Patterns

From the parses generated by the Charniak parser, we manually identified semantic role patterns for each of the 18 skill verbs[5]. For example, the patterns extracted for the skill verb "Advise" are:

Advise [Theme] (*for* [Purpose])
Advise (technically) *on*/*about* [Theme] (*for* [Purpose])

[5] "Architect" is not parsed as verb in the Charniak parser.

Advise clients/customers/employees/users *on*/*regarding* [Theme]

The corpus also contains embedded sub-semantic-role patterns, for example, for the "Theme" role we extracted the following sub-patterns:

(application*)* knowledge of/for [Concept]
sales of [Concept]
(technical) implementation of [Concept]

We have extracted a total of 74 such semantic role patterns from the skill titles.

3.6 Semantic Similarities between Skill Verbs

After examining the data, we also felt that a similarity metric between skill verbs and between matching role fillers would boost performance. Our initial attempt at this matching process was to match the skill verbs.

Many approaches to the problem of word/concept similarities are based on taxonomies, e.g., WordNet. The simplest approach is to count the number of nodes on the shortest path between two concepts in the taxonomy (Quillian, 1972). The fewer nodes on the path, the more similar the two concepts are. Despite its simplicity, this approach has achieved good results for some information retrieval (IR) tasks (Rada et al., 1989; Jarmasz & Szpakowicz, 2003). The assumption for this shortest path approach is that the links in the taxonomy represent uniform distances. However, in most taxonomies, sibling concepts deep in the taxonomy are usually more closely related than those higher up. Different approaches have been proposed to discount the depth of the concepts to overcome the problem. Budanitsky and Hirst (2006) thoroughly evaluated six of the approaches (i.e., Hirst & St-Onge, Leacock & Chodorow, Jiang & Conrath, Lin, Resnik, Wu & Palmer), and found out Jiang & Conrath (1997) was superior to the other approaches based on their evaluation experiments.

For our task, we compare two approaches to computing skill verb similarities: shortest path vs. Jiang & Conrath. Since the words are compared based on their specific senses, we first manually assign one appropriate sense for each of the 18 skill verbs from WordNet. We then use the implementation by Pedersen et al. (2004) to compute their similarity scores by both approaches.

Table 2 and 3 show the top nine pairs of skill verbs with the highest similarity scores from the two approaches. We can see that the two approaches agree on the top four pairs, but disagree on the rest in the list. One intuitive example is the pair "Lead" and "Manage" which is ranked the 5th by the Jiang & Conrath approach but ranked the 46th by the shortest path approach. It seems the Jiang & Conrath approach matches better with our human intuition for this example. While we are unable to compare these results with human performance, in general most of the similar skill verb pairs listed in the table don't look very similar for our domain. This may due to that WordNet is a general-purpose taxonomy -- although we have already selected the most appropriate sense for each verb, their relationship represented in the taxonomy may still be quite different from the relationship in our domain. A domain-specific taxonomy for skill verbs may improve the performance.

Shortest Path	
Apply	Use
Design	Plan
Apply	Implement
Implement	Use
Analyze	Apply
Analyze	Perform
Analyze	Support
Analyze	Use
Perform	Support
…	

Table 2. Shortest Path Results

Jiang & Conrath	
Apply	Use
Design	Plan
Apply	Implement
Implement	Use
Lead	**Manage**
Apply	Support
Support	Use
Apply	Sell
Sell	Use
…	

Table 3. Jiang & Conrath Results

Because the results were poor, we did not include verb similarity when evaluating our skill matching experiments.

4. Evaluation

In order to evaluate our approach to semantic similarity computation of skill descriptions, we first conducted experiments to evaluate how humans agree on this task, providing us with an upper bound accuracy for the task.

4.1 Inter-Rater Agreement and Upper Bound Accuracy

To assess inter-rater agreement, we randomly selected 75 skill pairs that share the same job role, or same secondary or primary job category, or from across the entire IBM expertise taxonomy.

These 75 skill pairs are then given to three raters to independently judge their similarities on a 5 point scale -- 1 as very similar, 2 as similar, 3 as neither similar nor dissimilar, 4 as dissimilar, and 5 as very dissimilar.

Since this 5 point scale is very fine-grained, we also convert the judgments to more coarse-grained, i.e., similar or not -- if it's 1 or 2, it's similar, otherwise, not similar.

The metric we used is the kappa statistic (Cohen, 1960; Krippendorff, 1980; Carletta, 1996), which factors out the agreement that is expected by chance:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where $P(A)$ is the observed agreement among the annotators, and $P(E)$ is the expected agreement, which is the probability that the annotators agree by chance.

Since the judgment on the 5 point scale is ordinal data, the weighted kappa statistic (Landis and Koch, 1977) is used to take the distance of disagreement into consideration (e.g., the disagreement between 1 and 2 is smaller than that between 1 and 5).

The inter-rater agreement results for both the fine-grained and coarse-grained judgments are shown in Table 4. In general, a kappa value above 0.80 represents perfect agreement, 0.60-0.80 represents significant agreement, 0.40-0.60 represents moderate agreement, and 0.20-0.40 is fair agreement. We can see that the agreement on the fine-grained judgment is moderate, whereas the agreement on the coarse-grained judgment significant.

	Fine-Grained Agreement	Coarse-Grained (Binary) Agreement
Kappa Statistic	**0.412**	**0.602**

Table 4. Inter-Rater Agreement Results.

From the inter-rater agreement evaluation, we can also get an *upper bound accuracy* for our task, i.e., human agreement without factoring out the agreement expected by chance (i.e., $P(A)$ in the kappa statistic). For our task, the average $P(A)$ for the coarse-grained (binary) judgment is 0.81, i.e., the upper bound accuracy for our task is 81%.

4.2 Evaluation of the Statistical Approach

We use the 75 skill pairs as test data to evaluate our semantic similarity approach against human judgments. Considering the reliability of the data, only the *coarse-grained (binary) judgments* are used. The gold standard is obtained by majority voting from the three raters, i.e., for a given skill pair, if two or more raters judge it as similar, then the gold standard answer is ‘similar’, otherwise it is ‘dissimilar’.

We first evaluated the standard statistical approach described in Section 3.1. Among 75 skill pairs, 53 of them were rated correctly according to the human judgments, i.e., 70.67% accuracy.

The error analysis shows that the many of the errors can be corrected if the skills are matched on their corresponding semantic roles. We will then evaluate the utility of the extracted semantic role information using a rule-based approach, and see whether it can improve the performance.

4.3 Evaluation of Semantic Role Matching for Skill Similarity Computation

For simplicity, we will only evaluate semantic role matching on their most important role, i.e. the "concept" role that specifies the key component of the skills, as introduced in Section 3.2.

There are at least two straightforward ways of performing semantic role matching for the skill similarity computation:

1) Match on the entire semantic role.
2) Match on their head nouns only.

However, both ways have their drawbacks:

1) Match on the entire semantic role.
It's a too strict matching criterion. It will miss many similar ones, for example,

Advise on [PeopleSoft CRM Field Service]
Apply [Siebel Field Services]

2) Match on their head nouns only.
It may not only miss the similar ones, for example,

Perform [Web Services *Planning*][6]	(head noun: planning)
Perform [Web Services *Assessment*]	(head noun: assessment)

but also misclassify the dissimilar ones as similar, for example,

Advise about [Async Transfer Mode (ATM) *Solutions*]	(head noun: solutions)
Advise about [CTI *Solutions*]	(head noun: solutions)

In order to solve these problems, we used a simple matching criterion from Tversky (1977): use only the common features for determining similarity. The similarity of two texts t_1 and t_2 is determined by:

$$\text{Similarity}(t_1, t_2) = \frac{2 \times (\#\,\text{common features between } t_1 \text{ and } t_2)}{\#\,\text{total features in } t_1 \text{ and } t_2}$$

This equation states that *two texts are similar if shared features are a large percentage of the total features.* We set a threshold of 0.5, requiring that 50% of the features be shared. We apply this criterion to only the text contained in the most important semantic role (concept).

The words in the calculation are preprocessed first: abbreviations are expanded, stop-words are excluded (e.g., "the", "a", "of" don't count as shared words), and the remaining words are stemmed (e.g., “manager” and “management” are counted as shared words), as was done in our previous information-theoretic approach. Words connected by punctuation (e.g., e-business, CA-11, software/hardware) are treated as separate words.

[6] The “concept” role is identified with brackets.

For this example,

Perform [*Web Services* Planning]
Perform [*Web Services* Assessment]

the shared words between the two "concept" roles are "Web" and "Services". The shared percentage is (2/3 + 2/3)/2 = 67.7% > 50%, so they are rated as similar.

Here is another example:

Advise on [*Field*/*Force* Management] for Telecom
Apply Knowledge of [Basic *Field Force* Automation]

The shared words between the two "concept" roles (bracketed) are "Field" & "Force", and the average shared percentage is (2/3 + 2/4)/2 = 58.3% > 50%, so they are similar.

We have also evaluated this approach on our test set with the 75 skill pairs. Among 75 skill pairs, 60 of them were rated correctly (i.e., 80% accuracy), which is very close to the upper bound accuracy, i.e., human agreement (81%).

The difference between this approach and Lin’s information content approach is that this computation is local and rule-based -- no corpus statistics are used, and using this approach it is also easier to set an intuitive threshold (e.g., 50%) for a classification problem (e.g., *similar or not* for our task). Lin's approach is suitable for computing a ranked list of similar pairs.

However, using this approach, there are still cases that can’t be classified correctly as similar if the surface words in the skill titles are not shared but the words are semantically similar, for example,

Advise [Business Value of IBM Software]
Identify [Customer Requirements and Product Usage]

More domain and commonsense knowledge would be needed to find the similarities between these two skills, e.g., “software” is a kind of “product” and “customer requirements” is indirectly related to “business value”. Although WordNet is a poplular resource for the noun similarity computation, there are many domain-specific terms and acronyms in our data that are not in WordNet, so a domain ontology may be needed for such approximate matches.

There are also cases that are mistagged as similar, for example,

Apply Knowledge of [Basic Field Force Automation]
Advise on [Sales Force Automation]

Although “Field Force Automation” and “Sales Force Automation” seem similar based upon their surface form, they are two quite different concepts. Again, more domain knowledge would be needed to distinguish such cases.

5. Conclusion

In this paper, we have presented our work on a semantic similarity computation for skill descriptions in natural language. We compared and evaluated four different natural language parsers for our task, and matched skills on their corresponding semantic roles extracted from the parses generated by one of these parsers. The evaluation results showed that the skill similarity computation based on semantic role matching can outperform a standard statistical approach and reach the level of human agreement.

Acknowledgements

The majority of this work was performed while Feng Pan was a summer intern at IBM T J Watson Research Center in Hawthorne, NY. Thanks to Yael Ravin and Jennifer Lai for supporting this work, Brian White for his help on the software, Michael McCord for assistance with the IBM ESG parser, and the IBM Expertise Taxonomy team for letting us use their data.

References

A. Budanitsky and G. Hirst. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*. 32(1):13-47.

J. Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254.

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of NAACL*.

J. Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 41, 687–699.

D. Gildea and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

A. Giuglea and A. Moschitti. 2006. Semantic Role Labeling via FrameNet, VerbNet and PropBank. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL),* Sydney, Australia

M. Jarmasz and S. Szpakowicz. 2003. Roget's Thesaurus and semantic similarity. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-2003)*, Borovetz, Bulgaria, pages 212–219, September.

J. J. Jiang and D. W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics (ROCLING X)*, Taiwan, pages 19–33.

C. Johnson, M. Petruck, C. Baker, M. Ellsworth, J. Ruppenhofer, and C. Fillmore. 2003. *Framenet: Theory and practice*. Berkeley, California.

D. Klein and C. D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL).*

K. Krippendorf. 1980. *Content Analysis: An introduction to its methodology.* Sage Publications.

R. J. Landis and G. G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics* 1977;33:159-74.

D. Lin. 1998a. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98).*

D. Lin. 1998b. Dependency-based evaluation of MINIPAR. In *Proceedings of the Workshop at LREC'98 on The Evaluation of Parsing Systems*, Granada, Spain.

M. McCord. 1989. Slot grammar: a system for simple construction of practical natural language grammars. *Natural Language and Logic*, pages 118-145.

G. A. Miller. 1990. WordNet: an On-line Lexical Database. *International Journal of Lexicography 3(4).*

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. Computational Linguistics, 31(1).

T. Pedersen, S. Patwardhan, and J. Michelizzi. 2004. Wordnet::similarity - measuring the relatedness of concepts. In *Proceedings of AAAI,* Intelligent Systems Demonstration, San Jose, CA.

S. Pradhan, W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky. 2004. Shallow Semantic Parsing using Support Vector Machines. In *Proceedings of the Human Language*

Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting (HLT/NAACL), Boston, MA.

S. Pradhan, W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky. 2005. Semantic role labeling using different syntactic views. In *Proceedings of the Association for Computational Linguistics 43rd annual meeting (ACL)*, Ann Arbor, MI.

M. R. Quillian. 1972. Semantic Memory, Semantic Information Processing. *Semantic information processing*, Cambridge, 1972

R. Rada, H. Mili, E. Bicknell, and M. Blettner. 1989. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30.

A. Tversky, Features of Similarity, *Psychological Review*, vol. 84, no. 4, pages 327-352, 1977.

N. Xue and M. Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Barcelona, Spain.