

# Mining Supervisor Evaluation and Peer Feedback in Performance Appraisals

Girish Keshav Palshikar, Sachin Pawar, Saheb Chourasia, Nitin Ramrakhiyani

TCS Research, Tata Consultancy Services Limited,  
54B Hadapsar Industrial Estate, Pune 411013, India.  
{gk.palshikar, sachin7.p, saheb.c, nitin.ramrakhiyani}@tcs.com

**Abstract.** Performance appraisal (PA) is an important HR process to periodically measure and evaluate every employee’s performance vis-a-vis the goals established by the organization. A PA process involves purposeful multi-step multi-modal communication between employees, their supervisors and their peers, such as self-appraisal, supervisor assessment and peer feedback. Analysis of the structured data and text produced in PA is crucial for measuring the quality of appraisals and tracking actual improvements. In this paper, we apply text mining techniques to produce insights from PA text. First, we perform sentence classification to identify strengths, weaknesses and suggestions of improvements found in the supervisor assessments and then use clustering to discover broad categories among them. Next we use multi-class multi-label classification techniques to match supervisor assessments to predefined broad perspectives on performance. Finally, we propose a short-text summarization technique to produce a summary of peer feedback comments for a given employee and compare it with manual summaries. All techniques are illustrated using a real-life dataset of supervisor assessment and peer feedback text produced during the PA of 4528 employees in a large multi-national IT company.

## 1 Introduction

Performance appraisal (PA) is an important HR process, particularly for modern organizations that crucially depend on the skills and expertise of their workforce. The PA process enables an organization to periodically measure and evaluate every employee’s performance. It also provides a mechanism to link the goals established by the organization to its each employee’s day-to-day activities and performance. Design and analysis of PA processes is a lively area of research within the HR community [13], [22], [10], [20].

The PA process in any modern organization is nowadays implemented and tracked through an IT system (the *PA system*) that records the interactions that happen in various steps. Availability of this data in a computer-readable database opens up opportunities to analyze it using automated statistical, data-mining and text-mining techniques, to generate novel and actionable insights / patterns and to help in improving the quality and effectiveness of the PA process [15], [19], [1]. Automated analysis of large-scale PA data is now facilitated

by technological and algorithmic advances, and is becoming essential for large organizations containing thousands of geographically distributed employees handling a wide variety of roles and tasks.

A typical PA process involves purposeful multi-step multi-modal communication between employees, their supervisors and their peers. In most PA processes, the communication includes the following steps: (i) in *self-appraisal*, an employee records his/her achievements, activities, tasks handled etc.; (ii) in *supervisor assessment*, the supervisor provides the criticism, evaluation and suggestions for improvement of performance etc.; and (iii) in *peer feedback* (aka *360° view*), the peers of the employee provide their feedback. There are several business questions that managers are interested in. Examples:

1. For my workforce, what are the broad categories of strengths, weaknesses and suggestions of improvements found in the supervisor assessments?
2. For my workforce, how many supervisor comments are present for each of a given fixed set of perspectives (which we call *attributes*), such as FUNCTIONAL\_EXCELLENCE, CUSTOMER\_FOCUS, BUILDING\_EFFECTIVE\_TEAMS etc.?
3. What is the summary of the peer feedback for a given employee?

In this paper, we develop text mining techniques that can automatically produce answers to these questions. Since the intended users are HR executives, ideally, the techniques should work with minimum training data and experimentation with parameter setting. These techniques have been implemented and are being used in a PA system in a large multi-national IT company.

The rest of the paper is organized as follows. Section 2 summarizes related work. Section 3 summarizes the PA dataset used in this paper. Section 4 applies sentence classification algorithms to automatically discover three important classes of sentences in the PA corpus viz., sentences that discuss strengths, weaknesses of employees and contain suggestions for improving her performance. Section 5 considers the problem of mapping the actual targets mentioned in strengths, weaknesses and suggestions to a fixed set of attributes. In Section 6, we discuss how the feedback from peers for a particular employee can be summarized. In Section 7 we draw conclusions and identify some further work.

## 2 Related Work

We first review some work related to sentence classification. Semantically classifying sentences (based on the sentence’s purpose) is a much harder task, and is gaining increasing attention from linguists and NLP researchers. McKnight and Srinivasan [12] and Yamamoto and Takagi [23] used SVM to classify sentences in biomedical abstracts into classes such as INTRODUCTION, BACKGROUND, PURPOSE, METHOD, RESULT, CONCLUSION. Cohen et al. [3] applied SVM and other techniques to learn classifiers for sentences in emails into classes, which are speech acts defined by a verb-noun pair, with verbs such as `request`, `propose`, `amend`, `commit`, `deliver` and nouns such as `meeting`, `document`, `committee`; see also [2].

Khoo et al. [9] uses various classifiers to classify sentences in emails into classes such as APOLOGY, INSTRUCTION, QUESTION, REQUEST, SALUTATION, STATEMENT, SUGGESTION, THANKING etc. Qadir and Riloff [17] proposes several filters and classifiers to classify sentences on message boards (community QA systems) into 4 speech acts: COMMISSIVE (speaker commits to a future action), DIRECTIVE (speaker expects listener to take some action), EXPRESSIVE (speaker expresses his or her psychological state to the listener), REPRESENTATIVE (represents the speaker’s belief of something). Hachey and Grover [7] used SVM and maximum entropy classifiers to classify sentences in legal documents into classes such as FACT, PROCEEDINGS, BACKGROUND, FRAMING, DISPOSAL; see also [18]. Deshpande et al. [5] proposes unsupervised linguistic patterns to classify sentences into classes SUGGESTION, COMPLAINT.

There is much work on a closely related problem viz., classifying sentences in dialogues through dialogue-specific categories called *dialogue acts* [21], which we will not review here. Just as one example, Cotterill [4] classifies questions in emails into the dialogue acts of YES.NO\_QUESTION, WH\_QUESTION, ACTION\_REQUEST, RHETORICAL, MULTIPLE\_CHOICE etc.

We could not find much work related to mining of performance appraisals data. Pawar et al. [16] uses kernel-based classification to classify sentences in both performance appraisal text and product reviews into classes SUGGESTION, APPRECIATION, COMPLAINT. Apte et al. [1] provides two algorithms for matching the descriptions of goals or tasks assigned to employees to a standard template of model goals. One algorithm is based on the co-training framework and uses goal descriptions and self-appraisal comments as two separate perspectives. The second approach uses semantic similarity under a weak supervision framework. Ramrakhiyani et al. [19] proposes label propagation algorithms to discover aspects in supervisor assessments in performance appraisals, where an aspect is modelled as a verb-noun pair (e.g. `conduct training`, `improve coding`).

### 3 Dataset

In this paper, we used the supervisor assessment and peer feedback text produced during the performance appraisal of 4528 employees in a large multi-national IT company. The corpus of supervisor assessment has 26972 sentences. The summary statistics about the number of words in a sentence is: min:4 max:217 average:15.5 STDEV:9.2 Q1:9 Q2:14 Q3:19.

### 4 Sentence Classification

The PA corpus contains several classes of sentences that are of interest. In this paper, we focus on three important classes of sentences viz., sentences that discuss strengths (class **STRENGTH**), weaknesses of employees (class **WEAKNESS**) and suggestions for improving her performance (class **SUGGESTION**). The strengths or weaknesses are mostly about the performance in work carried out, but sometimes they can be about the working style or other personal

qualities. The classes WEAKNESS and SUGGESTION are somewhat overlapping; e.g., a suggestion may address a perceived weakness. Following are two example sentences in each class.

#### STRENGTH:

- Excellent technology leadership and delivery capabilities along with ability to groom technology champions within the team.
- He can drive team to achieve results and can take pressure.

#### WEAKNESS:

- Sometimes exhibits the quality that he knows more than the others in the room which puts off others.
- Tends to stretch himself and team a bit too hard.

#### SUGGESTION:

- X has to attune himself to the vision of the business unit and its goals a little more than what is being currently exhibited.
- Need to improve on business development skills, articulation of business and solution benefits.

Several linguistic aspects of these classes of sentences are apparent. The subject is implicit in many sentences. The strengths are often mentioned as either noun phrases (NP) with positive adjectives (*Excellent technology leadership*) or positive nouns (*engineering strength*) or through verbs with positive polarity (*dedicated*) or as verb phrases containing positive adjectives (*delivers innovative solutions*). Similarly for weaknesses, where negation is more frequently used (*presentations are not his forte*), or alternatively, the polarities of verbs (*avoid*) or adjectives (*poor*) tend to be negative. However, sometimes the form of both the strengths and weaknesses is the same, typically a stand-alone sentiment-neutral NP, making it difficult to distinguish between them; e.g., *adherence to timing* or *timely closure*. Suggestions often have an imperative mood and contain secondary verbs such as *need to*, *should*, *has to*. Suggestions are sometimes expressed using comparatives (*better process compliance*). We built a simple set of patterns for each of the 3 classes on the POS-tagged form of the sentences. We use each set of these patterns as an unsupervised sentence classifier for that class. If a particular sentence matched with patterns for multiple classes, then we have simple tie-breaking rules for picking the final class. The pattern for the STRENGTH class looks for the presence of positive words / phrases like *takes ownership*, *excellent*, *hard working*, *commitment*, etc. Similarly, the pattern for the WEAKNESS class looks for the presence of negative words / phrases like *lacking*, *diffident*, *slow learner*, *less focused*, etc. The SUGGESTION pattern not only looks for keywords like *should*, *needs to* but also for POS based pattern like “a verb in the base form (VB) in the beginning of a sentence”.

We randomly selected 2000 sentences from the supervisor assessment corpus and manually tagged them (dataset D1). This labelled dataset contained 705, 103, 822 and 370 sentences having the class labels **STRENGTH**, **WEAKNESS**, **SUGGESTION** or **OTHER** respectively. We trained several multi-class classifiers on this dataset. Table 1 shows the results of 5-fold cross-validation experiments on dataset D1. For the first 5 classifiers, we used their implementation from the SciKit Learn library in Python ([scikit-learn.org](http://scikit-learn.org)). The features used for these classifiers were simply the sentence words along with their frequencies. For the last 2 classifiers (in Table 1), we used our own implementation. The overall *accuracy* for a classifier is defined as  $A = \frac{\#correct\_predictions}{\#data\_points}$ , where the denominator is 2000 for dataset D1. Note that the pattern-based approach is unsupervised i.e., it did not use any training data. Hence, the results shown for it are for the entire dataset and not based on cross-validation.

**Table 1.** Results of 5-fold cross validation for sentence classification on dataset D1.

Classifier	STRENGTH			WEAKNESS			SUGGESTION			A
	P	R	F	P	R	F	P	R	F	
Logistic Regression	0.715	0.759	0.736	0.309	0.204	0.246	0.788	0.749	0.768	0.674
Multinomial Naive Bayes	0.719	0.723	0.721	0.246	0.155	0.190	0.672	0.790	0.723	0.646
Random Forest	0.681	0.688	0.685	0.286	0.039	0.068	0.730	0.734	0.732	0.638
AdaBoost	0.522	0.888	0.657	0.265	0.087	0.131	0.825	0.618	0.707	0.604
Linear SVM	0.718	0.698	0.708	0.357	0.194	0.252	0.744	0.759	0.751	0.651
SVM with ADWSK [16]	0.789	0.847	<b>0.817</b>	0.491	0.262	0.342	0.844	0.871	<b>0.857</b>	<b>0.771</b>
Pattern-based	0.825	0.687	0.749	0.976	0.494	<b>0.656</b>	0.835	0.828	0.832	0.698

#### 4.1 Comparison with Sentiment Analyzer

We also explored whether a sentiment analyzer can be used as a baseline for identifying the class labels **STRENGTH** and **WEAKNESS**. We used an implementation of sentiment analyzer from TextBlob<sup>1</sup> to get a polarity score for each sentence. Table 2 shows the distribution of positive, negative and neutral sentiments across the 3 class labels **STRENGTH**, **WEAKNESS** and **SUGGESTION**. It can be observed that distribution of positive and negative sentiments is almost similar in **STRENGTH** as well as **SUGGESTION** sentences, hence we can conclude that the information about sentiments is not much useful for our classification problem.

#### 4.2 Discovering Clusters within Sentence Classes

After identifying sentences in each class, we can now answer question (1) in Section 1. From 12742 sentences predicted to have label **STRENGTH**, we extract

<sup>1</sup> <https://textblob.readthedocs.io/en/dev/>

**Table 2.** Results of TextBlob sentiment analyzer on the dataset D1

Sentence Class	Positive	Negative	Neutral
STRENGTH	544	44	117
WEAKNESS	44	24	35
SUGGESTION	430	52	340

**Table 3.** 5 representative clusters in strengths.

Strength cluster	Count
motivation expertise knowledge talent skill	1851
coaching team coach	1787
professional career job work working training practice	1531
opportunity focus attention success future potential impact result change	1431
sales retail company business industry marketing product	1251

nouns that indicate the actual strength, and cluster them using a simple clustering algorithm which uses the cosine similarity between word embeddings<sup>2</sup> of these nouns. We repeat this for the 9160 sentences with predicted label **WEAKNESS** or **SUGGESTION** as a single class. Tables 3 and 4 show a few representative clusters in strengths and in weaknesses, respectively. We also explored clustering 12742 **STRENGTH** sentences directly using CLUTO [8] and Carrot2 Lingo [14] clustering algorithms. Carrot2 Lingo<sup>3</sup> discovered 167 clusters and also assigned labels to these clusters. We then generated 167 clusters using CLUTO as well. CLUTO does not generate cluster labels automatically, hence we used 5 most frequent words within the cluster as its labels. Table 5 shows the largest 5 clusters by both the algorithms. It was observed that the clusters created by CLUTO were more meaningful and informative as compared to those by Carrot2 Lingo. Also, it was observed that there is some correspondence between noun clusters and sentence clusters. E.g. the nouns cluster **motivation expertise knowledge talent skill** (Table 3) corresponds to the CLUTO sentence cluster **skill customer management knowledge team** (Table 5). But overall, users found the nouns clusters to be more meaningful than the sentence clusters.

<sup>2</sup> We used 100 dimensional word vectors trained on Wikipedia 2014 and Gigaword 5 corpus, available at: <https://nlp.stanford.edu/projects/glove/>

<sup>3</sup> We used the default parameter settings for Carrot2 Lingo algorithm as mentioned at: <http://download.carrot2.org/head/manual/index.html>

**Table 4.** 5 representative clusters in weaknesses and suggestions.

Weakness cluster	Count
motivation expertise knowledge talent skill	1308
market sales retail corporate marketing commercial industry business	1165
awareness emphasis focus	1165
coaching team coach	1149
job work working task planning	1074

**Table 5.** Largest 5 sentence clusters within 12742 STRENGTH sentences

Algorithm	Cluster	#Sentences
CLUTO	performance performer perform years team	510
	skill customer management knowledge team	325
	role delivery work place show	289
	delivery manage management manager customer	259
	knowledge customer business experience work	250
Carrot2	manager manage	1824
	team team	1756
	delivery management	451
	manage team	376
	customer management	321

## 5 PA along Attributes

In many organizations, PA is done from a predefined set of perspectives, which we call *attributes*. Each attribute covers one specific aspect of the work done by the employees. This has the advantage that we can easily compare the performance of any two employees (or groups of employees) along any given attribute. We can correlate various performance attributes and find dependencies among them. We can also cluster employees in the workforce using their supervisor ratings for each attribute to discover interesting insights into the workforce. The HR managers in the organization considered in this paper have defined 15 attributes (Table 6). Each attribute is essentially a work item or work category described at an abstract level. For example, FUNCTIONAL\_EXCELLENCE covers any tasks, goals or activities related to the software engineering life-cycle (e.g., requirements analysis, design, coding, testing etc.) as well as technologies such as databases, web services and GUI.

In the example in Section 4, the first sentence (which has class STRENGTH) can be mapped to two attributes: FUNCTIONAL\_EXCELLENCE and BUILDING\_EFFECTIVE\_TEAMS. Similarly, the third sentence (which has class WEAKNESS) can be mapped to the attribute INTERPERSONAL\_EFFECTIVENESS and so forth. Thus, in order to answer the second question in Section 1, we need to map each sentence in each of the 3 classes to zero, one, two or more attributes, which is a multi-class multi-label classification problem.

We manually tagged the same 2000 sentences in Dataset D1 with attributes, where each sentence may get 0, 1, 2, etc. up to 15 class labels (this is dataset D2). This labelled dataset contained 749, 206, 289, 207, 91, 223, 191, 144, 103, 80, 82, 42, 29, 15, 24 sentences having the class labels listed in Table 6 in the same order. The number of sentences having 0, 1, 2, or more than 2 attributes are: 321, 1070, 470 and 139 respectively. We trained several multi-class multi-label classifiers on this dataset. Table 7 shows the results of 5-fold cross-validation experiments on dataset D2.

Precision, Recall and F-measure for this multi-label classification are computed using a strategy similar to the one described in [6]. Let  $P_i$  be the set of

**Table 6.** Strengths, Weaknesses and Suggestions along Performance Attributes

Performance Attributes	#Strengths	#Weaknesses	#Suggestions
FUNCTIONAL_EXCELLENCE	321	26	284
BUILDING_EFFECTIVE_TEAMS	80	6	89
INTERPERSONAL_EFFECTIVENESS	151	16	97
CUSTOMER_FOCUS	100	5	76
INNOVATION_MANAGEMENT	22	4	53
EFFECTIVE_COMMUNICATION	53	17	124
BUSINESS_ACUMEN	39	10	103
TAKING_OWNERSHIP	47	3	81
PEOPLE_DEVELOPMENT	31	8	57
DRIVE_FOR_RESULTS	37	4	30
STRATEGIC_CAPABILITY	8	4	51
WITHSTANDING_PRESSURE	16	6	16
DEALING_WITH_AMBIGUITIES	4	8	12
MANAGING_VISION_AND_PURPOSE	3	0	9
TIMELY_DECISION_MAKING	6	2	10

**Table 7.** Results of 5-fold cross validation for multi-class multi-label classification on dataset D2.

Classifier	Precision P	Recall R	F
Logistic Regression	0.715	0.711	<b>0.713</b>
Multinomial Naive Bayes	0.664	0.588	0.624
Random Forest	0.837	0.441	0.578
AdaBoost	0.794	0.595	0.680
Linear SVM	0.722	0.672	0.696
Pattern-based	0.750	0.679	<b>0.713</b>

predicted labels and  $A_i$  be the set of actual labels for the  $i^{th}$  instance. Precision and recall for this instance are computed as follows:

$$Precision_i = \frac{|P_i \cap A_i|}{|P_i|}, \quad Recall_i = \frac{|P_i \cap A_i|}{|A_i|}$$

It can be observed that  $Precision_i$  would be undefined if  $P_i$  is empty and similarly  $Recall_i$  would be undefined when  $A_i$  is empty. Hence, overall precision and recall are computed by averaging over all the instances except where they are undefined. Instance-level F-measure can not be computed for instances where either precision or recall are undefined. Therefore, overall F-measure is computed using the overall precision and recall.

## 6 Summarization of Peer Feedback using ILP

The PA system includes a set of peer feedback comments for each employee. To answer the third question in Section 1, we need to create a summary of all the



peer feedback comments about a given employee. As an example, following are the feedback comments from 5 peers of an employee.

1. vast knowledge on different technologies
2. His experience and vast knowledge mixed with his positive attitude, willingness to teach and listen and his humble nature.
3. Approachable, Knowledgeable and is of helping nature.
4. Dedication, Technical expertise and always supportive
5. Effective communication and team player

The individual sentences in the comments written by each peer are first identified and then POS tags are assigned to each sentence. We hypothesize that a good summary of these multiple comments can be constructed by identifying a set of *important* text fragments or phrases. Initially, a set of candidate phrases is extracted from these comments and a subset of these candidate phrases is chosen as the final summary, using Integer Linear Programming (ILP). The details of the ILP formulation are shown in Table 8. As an example, following is the summary generated for the above 5 peer comments.

**humble nature, effective communication, technical expertise, always supportive, vast knowledge**

Following rules are used to identify candidate phrases:

- An adjective followed by **in** which is followed by a noun phrase (e.g. **good in customer relationship**)
- A verb followed by a noun phrase (e.g. **maintains work life balance**)
- A verb followed by a preposition which is followed by a noun phrase (e.g. **engage in discussion**)
- Only a noun phrase (e.g. **excellent listener**)
- Only an adjective (e.g. **supportive**)

Various parameters are used to evaluate a candidate phrase for its *importance*. A candidate phrase is more important:

- if it contains an adjective or a verb or its headword is a noun having WordNet lexical category *noun.attribute* (e.g. nouns such as **dedication, sincerity**)
- if it contains more number of words
- if it is included in comments of multiple peers
- if it represents any of the performance attributes such as *Innovation, Customer, Strategy* etc.

A complete list of parameters is described in detail in Table 8.

There is a trivial constraint  $C_0$  which makes sure that only  $K$  out of  $N$  candidate phrases are chosen. A suitable value of  $K$  is used for each employee depending on number of candidate phrases identified across all peers (see Algorithm 1). Another set of constraints ( $C_1$  to  $C_{10}$ ) make sure that at least one phrase is selected for each of the leadership attributes. The constraint  $C_{11}$  makes sure that multiple phrases sharing the same headword are not chosen at a time. Also,

single word candidate phrases are chosen only if they are adjectives or nouns with lexical category *noun.attribute*. This is imposed by the constraint  $C_{12}$ . It is important to note that all the constraints except  $C_0$  are soft constraints, i.e. there may be feasible solutions which do not satisfy some of these constraints. But each constraint which is not satisfied, results in a penalty through the use of slack variables. These constraints are described in detail in Table 8.

The objective function maximizes the total *importance* score of the selected candidate phrases. At the same time, it also minimizes the sum of all slack variables so that the minimum number of constraints are broken.

```

Data:  $N$ : No. of candidate phrases
Result:  $K$ : No. of phrases to select as part of summary
if  $N \leq 10$  then
  |  $K \leftarrow \lfloor N * 0.5 \rfloor$ ;
else if  $N \leq 20$  then
  |  $K \leftarrow \lfloor \text{getNoOfPhrasesToSelect}(10) + (N - 10) * 0.4 \rfloor$ ;
else if  $N \leq 30$  then
  |  $K \leftarrow \lfloor \text{getNoOfPhrasesToSelect}(20) + (N - 20) * 0.3 \rfloor$ ;
else if  $N \leq 50$  then
  |  $K \leftarrow \lfloor \text{getNoOfPhrasesToSelect}(30) + (N - 30) * 0.2 \rfloor$ ;
else
  |  $K \leftarrow \lfloor \text{getNoOfPhrasesToSelect}(50) + (N - 50) * 0.1 \rfloor$ ;
end
if  $K < 4$  and  $N \geq 4$  then
  |  $K \leftarrow 4$ 
else if  $K < 4$  then
  |  $K \leftarrow N$ 
else if  $K > 20$  then
  |  $K \leftarrow 20$ 
end

```

**Algorithm 1:** *getNoOfPhrasesToSelect* (For determining number of phrases to select to include in summary)

## 6.1 Evaluation of auto-generated summaries

We considered a dataset of 100 employees, where for each employee multiple peer comments were recorded. Also, for each employee, a manual summary was generated by an HR personnel. The summaries generated by our ILP-based approach were compared with the corresponding manual summaries using the ROUGE [11] unigram score. For comparing performance of our ILP-based summarization algorithm, we explored a few summarization algorithms provided by the Sumy package<sup>4</sup>. A common parameter which is required by all these

<sup>4</sup> <https://github.com/miso-belica/sumy>

**Table 8.** Integer Linear Program (ILP) formulation

<b>Parameters:</b> <ul style="list-style-type: none"> <li>– <math>N</math>: No. of phrases</li> <li>– <math>K</math>: No. of phrases to be chosen for inclusion in the final summary</li> <li>– <math>Freq</math>: Array of size <math>N</math>, <math>Freq_i</math> = no. of distinct peers mentioning the <math>i^{th}</math> phrase</li> <li>– <math>Adj</math>: Array of size <math>N</math>, <math>Adj_i = 1</math> if the <math>i^{th}</math> phrase contains any adjective</li> <li>– <math>Verb</math>: Array of size <math>N</math>, <math>Verb_i = 1</math> if the <math>i^{th}</math> phrase contains any verb</li> <li>– <math>NumWords</math>: Array of size <math>N</math>, <math>NumWords_i = 1</math> no. of words in the <math>i^{th}</math> phrase</li> <li>– <math>NounCat</math>: Array of size <math>N</math>, <math>NounCat_i = 1</math> if lexical category (WordNet) of headword of the <math>i^{th}</math> phrase is <i>noun.attribute</i></li> <li>– <math>InvalidSingleNoun</math>: Array of size <math>N</math>, <math>InvalidSingleNoun_i = 1</math> if the <math>i^{th}</math> phrase is single word phrase which is neither an adjective nor a noun having lexical category (WordNet) <i>noun.attribute</i></li> <li>– <math>Leadership, Team, Innovation, Communication, Knowledge, Delivery, Ownership, Customer, Strategy, Personal</math>: Indicator arrays of size <math>N</math> each, representing whether any phrase corresponds to a particular performance attribute, e.g. <math>Customer_i = 1</math> indicates that <math>i^{th}</math> phrase is of type <i>Customer</i></li> <li>– <math>S</math>: Matrix of dimensions <math>N \times N</math>, where <math>S_{ij} = 1</math> if headwords of <math>i^{th}</math> and <math>j^{th}</math> phrase are same</li> </ul>
<b>Variables:</b> <ul style="list-style-type: none"> <li>– <math>X</math>: Array of <math>N</math> <b>binary</b> variables, where <math>X_i = 1</math> only when <math>i^{th}</math> phrase is chosen to be the part of final summary</li> <li>– <math>S_1, S_2, \dots, S_{12}</math>: <b>Integer</b> slack variables</li> </ul>
<b>Objective:</b> Maximize $\sum_{i=1}^N ((NounCat_i + Adj_i + Verb_i + 1) \cdot Freq_i \cdot NumWords_i \cdot X_i) - 10000 \cdot \sum_{j=1}^{12} S_j$
<b>Constraints:</b> <p><math>C_0: \sum_{i=1}^N X_i = K</math> (Exactly <math>K</math> phrases should be chosen)</p> <p><math>C_1: \sum_{i=1}^N (Leadership_i \cdot X_i) + S_1 \geq 1</math></p> <p><math>C_2: \sum_{i=1}^N (Team_i \cdot X_i) + S_2 \geq 1</math></p> <p><math>C_3: \sum_{i=1}^N (Knowledge_i \cdot X_i) + S_3 \geq 1</math></p> <p><math>C_4: \sum_{i=1}^N (Delivery_i \cdot X_i) + S_4 \geq 1</math></p> <p><math>C_5: \sum_{i=1}^N (Ownership_i \cdot X_i) + S_5 \geq 1</math></p> <p><math>C_6: \sum_{i=1}^N (Innovation_i \cdot X_i) + S_6 \geq 1</math></p> <p><math>C_7: \sum_{i=1}^N (Communication_i \cdot X_i) + S_7 \geq 1</math></p> <p><math>C_8: \sum_{i=1}^N (Customer_i \cdot X_i) + S_8 \geq 1</math></p> <p><math>C_9: \sum_{i=1}^N (Strategy_i \cdot X_i) + S_9 \geq 1</math></p> <p><math>C_{10}: \sum_{i=1}^N (Personal_i \cdot X_i) + S_{10} \geq 1</math>  (At least one phrase should be chosen to represent each leadership attribute)</p> <p><math>C_{11}: \sum_{i=1}^N \sum_{j=1, s.t. i \neq j}^N (S_{ij} \cdot (X_i + X_j - 1)) + S_{11} \leq 0</math>  (No duplicate phrases should be chosen)</p> <p><math>C_{12}: \sum_{i=1}^N (InvalidSingleNoun_i \cdot X_i) - S_{12} \leq 0</math>  (Single word noun phrases are not preferred if they are not <i>noun.attribute</i>)</p>

algorithms is number of sentences keep in the final summary. ILP-based summarization requires a similar parameter  $K$ , which is automatically decided based on number of total candidate phrases. Assuming a sentence is equivalent to roughly 3 phrases, for Sumy algorithms, we set number of sentences parameter to the ceiling of  $K/3$ . Table 9 shows average and standard deviation of ROUGE unigram f1 scores for each algorithm, over the 100 summaries. The performance of ILP-based summarization is comparable with the other algorithms, as the two sample t-test does not show statistically significant difference. Also, human evaluators preferred phrase-based summary generated by our approach to the other sentence-based summaries.

**Table 9.** Comparative performance of various summarization algorithms

Algorithm	ROUGE unigram F1	
	Average	Std. Deviation
LSA	0.254	0.146
TextRank	0.254	0.146
LexRank	0.258	0.148
ILP-based summary	0.243	0.15

## 7 Conclusions and Further Work

In this paper, we presented an analysis of the text generated in Performance Appraisal (PA) process in a large multi-national IT company. We performed sentence classification to identify strengths, weaknesses and suggestions for improvements found in the supervisor assessments and then used clustering to discover broad categories among them. As this is non-topical classification, we found that SVM with ADWS kernel [16] produced the best results. We also used multi-class multi-label classification techniques to match supervisor assessments to predefined broad perspectives on performance. Logistic Regression classifier was observed to produce the best results for this topical classification. Finally, we proposed an ILP-based summarization technique to produce a summary of peer feedback comments for a given employee and compared it with manual summaries.

The PA process also generates much structured data, such as supervisor ratings. It is an interesting problem to compare and combine the insights from discovered from structured data and unstructured text. Also, we are planning to automatically discover any additional performance attributes to the list of 15 attributes currently used by HR.

## References

1. M. Apte, S. Pawar, S. Patil, S. Baskaran, A. Shrivastava, and G.K. Palshikar. Short text matching in performance management. In *Proceedings of the 21st International*

- Conference on Management of Data (COMAD 2016)*, pages 13–23, 2016.
2. Vitor R. Carvalho and William W. Cohen. Improving "email speech acts" analysis via n-gram selection. In *Proceedings of the HLT-NAACL 2006 Workshop on Analyzing Conversations in Text and Speech*, ACTS '09, pages 35–41, 2006.
  3. W.W. Cohen, V.R. Carvalho, and T.M. Mitchell. Learning to classify email into "speech acts". In *Proc. Empirical Methods in Natural Language Processing (EMNLP-2004)*, pages 309–316, 2004.
  4. Rachel Cotterill. Question classification for email. In *Proc. Ninth Int. Conf. Computational Semantics (IWCS 2011)*, 2011.
  5. S. Deshpande, G.K. Palshikar, and G. Athiappan. An unsupervised approach to sentence classification. In *Proc. Int. Conf. on Management of Data (COMAD 2010)*, pages 88–99, 2010.
  6. Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In *PAKDD 2004*, pages 22–30, 2004.
  7. B. Hachey and C. Grover. Sequence modelling for sentence classification in a legal summarisation system. In *Proc. 2005 ACM Symposium on Applied Computing*, 2005.
  8. George Karypis. Cluto-a clustering toolkit. Technical report, DTIC Document, 2002.
  9. A. Khoo, Y. Marom, and D. Albrecht. Experiments with sentence classification. In *Proc. 2006 Australasian Language Technology Workshop (ALTW2006)*, pages 18–25, 2006.
  10. P.E. Levy and J.R. Williams. The social context of performance appraisal: a review and framework for the future. *Journal of Management*, 30(6):881–905, 2004.
  11. Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004.
  12. L. McKnight and P. Srinivasan. Categorization of sentence types in medical abstracts. In *Proc. American Medical Informatics Association Annual Symposium*, pages 440–444, 2003.
  13. K.R. Murphy and J. Cleveland. *Understanding Performance Appraisal: Social, Organizational and Goal-Based Perspective*. Sage Publishers, 1995.
  14. Stanislaw Osinski, Jerzy Stefanowski, and Dawid Weiss. Lingo: Search results clustering algorithm based on singular value decomposition. In *Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'04 Conference held in Zakopane, Poland, May 17-20, 2004*, pages 359–368, 2004.
  15. G.K. Palshikar, S. Deshpande, and S. Bhat. Quest: Discovering insights from survey responses. In *Proceedings of the 8th Australasian Data Mining Conf. (AusDM09)*, pages 83–92, 2009.
  16. S. Pawar, N. Ramrakhiyani, G. K. Palshikar, and S. Hingmire. Deciphering review comments: Identifying suggestions, appreciations and complaints. In *Proc. 20th Int. Conf. on Applications of Natural Language to Information Systems (NLDB 2015)*, LNCS 9103, pages 204–211, 2015.
  17. Ashequl Qadir and Ellen Riloff. Classifying sentences as speech acts in message board posts. In *Proc. Empirical Methods in Natural Language Processing (EMNLP-2011)*, 2011.
  18. N. Ramrakhiyani, S. Pawar, and G.K. Palshikar. A system for classification of propositions of the indian supreme court judgements. In *Proc. 5th 2013 Forum on Information Retrieval Evaluation (FIRE 2013)*, pages 1–4, 2013.

19. N. Ramrakhiyani, S. Pawar, G.K. Palshikar, and M. Apte. Aspects from appraisals: A label propagation with prior induction approach. In *Proceedings of the 21st International Conference on Applications of Natural Language to Information Systems (NLDB 2016)*, volume LNCS 9612, pages 301–309, 2016.
20. M. Schraeder, J. Becton, and R. Portis. A critical examination of performance appraisals. *The Journal for Quality and Participation*, (spring):20–25, 2007.
21. Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3), 2000.
22. C. Viswesvaran. Assessment of individual job performance: a review of the past century and a look ahead. In N. Anderson, D.S. Ones, H.K. Sinangil, and C. Viswesvaran, editors, *Handbook of Industrial, Work and Organizational Psychology*. Sage Publishers, 2001.
23. Y. Yamamoto and T. Takagi. A sentence classification system for multi biomedical literature summarization. In *Proc. 21st International Conference on Data Engineering Workshops*, pages 1163–1168, 2005.