Mining special features to improve the performance of e-commerce product selection and resume processing

Abhishek Sainani*, P. Krishna Reddy and Sumit Maheshwari

Center for Data Engineering,
International Institute of Information Technology,
Gachibowli, Hyderabad 500 032, India
E-mail: abhisainani@gmail.com
E-mail: pkreddy@mail.iiit.ac.in
E-mail: khushisezindagi@gmail.com
*Corresponding author

Abstract: In the literature, research efforts are going on to extract interesting information from text documents to improve the performance of information-based services. Interesting information is extracted after identifying features from each document. In this paper, we have proposed the notion of 'special feature' which is a new kind of knowledge that can be used to improve the performance of information-based services. A feature is a special feature if only very few documents in the dataset possess it. Given a text document dataset, we have proposed a methodology to extract special features. By using the notion of special features, we have also proposed frameworks to improve the performance of product selection in the e-commerce environment and the process of resume selection. The experiment results on real datasets show that it is possible to improve the efficiency of the applications with the proposed approach.

Keywords: e-commerce; information extraction; resume processing; text mining; computational science and engineering; special features.

Reference to this paper should be made as follows: Sainani, A., Reddy, P.K. and Maheshwari, S. (2012) 'Mining special features to improve the performance of e-commerce product selection and resume processing', *Int. J. Computational Science and Engineering*, Vol. 7, No. 1, pp.82–95.

Biographical notes: Abhishek Sainani has completed BTech and MS by Research in Computer Science from International Institute of Information Technology, Hyderabad (IIIT-H), India. Currently, he is working as a Database Administrator in S&P Capital IQ, Hyderabad, India. His areas of interests are data mining and text mining.

P. Krishna Reddy has been a part of IIIT, Hyderabad (IIIT-H), India since 2002. He received his PhD in Computer Science from Jawaharlal Nehru University, New Delhi. He was a Researcher at Institute of Industrial Science, University of Tokyo during 1997–2002. His research interests include data mining, web mining, data management, transaction models, distributed computing, and ICTs for agriculture. He has published about 90 referred research papers which include 12 journal papers and three book chapters.

Sumit Maheshwari has completed BTech and MS by Research in Computer Science from IIIT, Hyderabad (IIIT-H), India. Currently, he is working as a Software Engineer in Informatica Business Solutions Pvt. Ltd., Bangalore, India. His areas of interests are data warehousing, data mining, and text mining.

This paper is a revised and expanded version of a paper entitled 'An approach to extract special skills to improve the performance of resume selection' presented at the 6th International Workshop on Databases in Networked Information Systems (DNIS) 2010, Aizu-Wakamatsu, Japan, 29–31 March 2010.

1 Introduction

The proliferation of the World Wide Web has led to the problem of information overload for internet users as more and more data is being added online on a regular basis. Various research efforts are going on in the field of data mining (Chu et al., 2011), information extraction (Carlson et al., 2010; Wu and Weld, 2010; Qiu and Yang, 2010; Crestan and Pantel, 2010), machine learning (Jin et al., 2009; Chau and Chen, 2008) and related fields to

automatically extract meaningful information from different kind of datasets.

Identification of features from text documents is an important step of information extraction or text mining process. For example, important words and phrases can be identified as features of the text document. In the literature, efforts are going on to investigate new ways of extracting and processing the extracted features for generating new kinds of knowledge to improve performance of information-based services. In Liu et al. (2001), the notion of unexpected information is presented using which one of the two similar web sites know the modification made by other web site by comparing and contrasting the corresponding web pages. Also, in case of outliers mining (Aggarwal and Yu, 2001), a document is considered as outlier if it contains many distinct features as compared to other documents.

In certain applications, a text document possesses some specialness, which is exhibited through few special features. In this paper we have introduced the notion of special features and proposed a method to organise these features. We explain the motivation through the following two examples.

- *Example 1:* Consider the case of product selection in an e-commerce environment. A customer wants to buy a Nokia mobile phone. Table 1 shows a sample of product features for N-77 and N-82 models of Nokia mobile phone (Mobile Phones, 2011). To select a mobile phone, the customer has to carefully go through the features of about 70 Nokia mobile phone models, where each mobile phone description consists of about 30 features. As different variants of mobile phones are available, the customer has to manually browse several web pages to identify the specialty of each mobile phone. Even after spending sufficient amount of time, sometimes, a customer may not feel confident of his/her choice. In the e-commerce environment, customers face similar problems in several scenarios such as selecting a laptop, television, iPod, and camera and so on. Every product has some specialness which is exhibited through one or few of its special features. There is an opportunity for performance improvement in this case, if we identify the special features and organise them in an efficient manner so that customers can quickly go through special features and other features to select the product.
- *Example 2:* In the internet era, large number of resumes are received on-line, through e-mails or through services provided by companies like Info Edge (India) Limited (2011). In case of resume extraction system, the resumes received in large numbers are reduced to few hundred potential ones based on some filtering techniques or search services. The set of resumes hence obtained are similar to each other as they satisfy the search criteria or requirements for a company. After getting the set of similar resumes, it is a difficult and time consuming process to manually analyse each resume to select appropriate resume. Normally, every resume contains some special features. If we identify special features and use the same for selecting the resumes, it is possible to improve the performance of resume selection process.

Table 1 Sample product features of N-77 and N-82 mobile phone separated by delimiter (,)

N-77 features	*N-82 features*
Network UMTS gsm 900 gsm 1,800 gsm 1,900, announced 2007 1q (February), weight 114 g display type TFT 16 m colours, display size 240 × 320 pixels 2.4 inches, ringtones type polyphonic (64 channels) mp3, vibration yes, phonebook yes, call records yes, card slot Microsd (transflash) hotswap, operating system Symbian OS 9.2 s60 rel 3.1, camera 2 mp 1,600 × 1,200 pixels video(cif)-flash secondary cif video call camera, GPRS/data speed class 11, messaging SMS MMS e-mail instant messaging, infrared port no games yes java downloadable, colours black 3g 384 kbps, Bluetooth v1.2 with a2dp DVB-H TV broadcast receiver, video calling push to talk, Java MIDP 2.0,mp3/m4a/aac/eaac+/wma player t9, stereo fm radio, voice command/dial pim including calendar to-do list and printing document viewer, photo/video editor	Network gsm 850 900 1,800 1,900 hsdpa, announced 2007 4q (November),weight 114 g, display type TFT 16m colours, display size 240 × 320 pixels 2.4 inches, ringtones type polyphonic monophonic true tones mp3, vibration yes, phonebook practically unlimited-entries and fields photocall, call records detailed max 30 days, camera 5 mp 2,592 × 1,944 pixels carl zeiss-optics autofocus video(vga 30fps), operating system Symbian OS 9.2 s60 rel 3.1, card slot Microsd hot swap 2gb card included, GPRS/data speed class 32 107 kbps, messaging SMS MMS e-mail instant messaging, infrared port no, games yes downloadable, WLAN Wi-Fi 802.11 b/g UPNP technology, Bluetooth v2.0 with a2dp, t9, USB v2.0 MicroUSB, browser wap 2.0/xhtml html, built-in gps receiver, motion sensor (with ui auto-rotate), Java MIDP 2.0, mp3/aac/aac+/eaac+/wma player

In this paper, we have shown that the notion of special features can be used to improve the performance of information-based services like product selection process in e-commerce system, and resume selection process. We have defined the notion of special feature and proposed a generalised approach to organise the same. In case of e-commerce environment, the proposed approach identifies special features of products and organises them in an efficient manner to ease the process of product selection. In case of resume selection process, the proposed approach identifies special skills from the resume and organises them in an efficient manner to ease the process of resume selection. We show the efficiency of the proposed approach by conducting experiments on real datasets.

We have been investigating for better feature processing methods to improve the performance of product selection and resume processing. In Maheshwari and Reddy (2009), we have proposed the notion of special features to improve the performance of product selection in e-commerce environment. In Maheshwari et al. (2010), we have extended the notion of special features to improve the performance of resume processing by processing the skills section of resume dataset. In this paper, we have improved the special feature organisation approach and the presentation.

The rest of the paper is organised as follows. In the next section we discuss the related work. In Section 3, we present the notion of special features, and feature organisation algorithm. In Section 4, we discuss the framework of the proposed approach on product selection framework. In Section 5, we discuss the application of our technique on resume selection framework. Section 6 concludes the paper.

2 Related work

In this section, we first discuss the related research efforts about the extraction of outliers and interesting information from different kinds of datasets. Next, we discuss the research efforts concerning to resume processing.

Statistical techniques assume that the given dataset has a distribution model. Outliers are those points that satisfy a discordancy test, that is, they are significantly far from their expected position, given the hypothesised distribution (Barnett and Lewis, 1994). The subgroup discovery task (Klosgen, 1996) aims at finding an interesting subgroup of objects with common characteristics with respect to a given attribute, called the target variable. In Knorr and Ng (1998), an approach has been proposed to mine the distance-based outliers. In Breunig et al. (2000), an approach has been proposed in which each object is assigned a degree of being an outlier, which is called local outlier factor. It is local in that the degree depends on how isolated the object is with respect to the surrounding neighbourhood. The notion of K-nearest neighbour has also been used to identify outliers in Ramaswamy et al. (2000). In Aggarwal and Yu (2001), anomalies are detected by searching for subspaces in which the data density is exceptionally lower than the mean density of the whole dataset. An abnormal lower projection is one in which the density of the data is exceptionally lower than the average. In Wei et al. (2003), a novel definition of outlier is proposed based on a hypergraph model for categorical data. In Zhu et al. (2005), an approach is proposed to find example-based outliers in a dataset. In Zhang and Wang (2006), an effort has been made to find the subspaces in which a given point is an outlier. The approach detects outliers with monotonic property: If a point is an outlier in a subspace, then it will be an outlier in any superset of this subspace. In Angiulli et al. (2009), a criterion is defined to measure the abnormality of subsets of attribute values featured by a given object with respect to a reference data population, and it discusses methods to evaluate it.

In web mining research area, efforts are being made in the literature to identify outliers in a given set of web documents. Web content outliers are documents with varying contents compared to similar web documents taken from the same domain. Web content outlier mining (Agyemang et al., 2004) involves finding outlier documents from a given set of documents. In Agyemang et al. (2005b), an approach is presented to mine outliers considering an n-gram matching using domain dictionary whereas in Agyemang et al. (2005a), an effort has been made to find web content outliers using n-grams without use of domain dictionary.

In Liu et al. (2001), an approach has been proposed to discover unexpected information from the competitors' site. It compares the users' web pages with those of the competitors and finds different kinds of unexpected information. Search engines and comparative web search systems have also emerged to help customers in seeking relevant content. Comparative web search systems (Sun et al., 2006) seek relevant and comparative information from the web to help customers conduct comparisons among a set of topics. Given a set of queries which represents the topics that a user wants to compare, the system automatic rank and retrieves the web pages, cluster the pages, and extracts representative key phrases. An effort has been made to extract the unique properties of any given record in a dataset (Paravastu et al., 2008). The approach determines what makes a given record unique or different from the majority of the records in a dataset.

The research efforts in the area of resume processing are as follows.

A cascaded two-pass information extraction framework is designed in Yu et al. (2005). In the first pass, the general information is extracted by segmenting the entire resume into consecutive blocks and each block is annotated with a label indicating its category. In the second pass, detailed information pieces are further extracted within the boundary of certain blocks. A four phase approach for resume processing is proposed in Karamatli and Akyokus (2010). In the first step, a resume is segmented into blocks according to their information types. In the second step, named entities are found by using special chunkers for each information type. In the third step, found named entities are clustered according to their distance in text and information type.

In the fourth step, normalisation methods are applied to the text. In the end, the extracted information is produced in JSON or XML format.

The work presented in this paper is different from the preceding approaches. The outlier algorithms discussed in data mining area deals with numerical datasets. The outlier algorithms proposed in web mining area concentrates more on classifying text document as an outlier. The approach to find uniqueness of a data record is based on predefined properties. In the proposed approach, an effort has been made to extract special properties of each object to improve the performance of information-based services. We have demonstrated that the effectiveness of proposed approach by considering recommendation system in e-commerce and resume processing applications.

3 Special features and organisation

We first explain the notion of special features. Next, we present the approach to organise the special features.

3.1 Special features and degree of specialness

We explain the proposed approach by considering a set of text documents. In general, the proposed approach can work on any dataset that is in the form of objects and their properties. We present the proposed approach by considering object as text-based documents and the properties as text-based features.

Consider a set of text documents. Each text document corresponds to an object in the mini universe, and is represented by a set of text-based features. For example, the textual description of mobile phone specifications is a text document, each specification of mobile phone is a feature.

We now define the term 'similar documents'. Two documents are similar if the number of common features between them is greater than a given threshold. For example, different variants of mobile phones form a group of similar products as they have common features.

Normally, each text document possesses specialty which is exhibited through one or few special features. Due to its special features, each text document can be distinguished from the other similar documents. The intuition here is that such special features of each document are helpful in building efficient information-based services. For example, in case of e-commerce environment, if we identify special features of each product and show it to the customer, the time taken by the customer to make a decision for selecting a product would be reduced in comparison to showing all the information about the products. Note that the number of special features is much smaller when compared to the total information about the product. So the customer can quickly browse through the special features of all the products to take the decision as compared to all the features for every product.

The main issue here is how to measure the specialness of features of all text documents and organise the features according to their specialness in an effective manner. For this, we introduce the notion of degree of specialness.

Let f_j be the j^{th} feature of o_i which is the i^{th} object, such that $f_j \in f(o_i)$ where $f(o_i)$ is the set of features of object o_i. The degree of specialness (DS) of f_j is its capability of making the object o_i, separate/distinct/unique/special from other objects. The DS value for a feature varies between zero to one (both inclusive). The DS value of the feature f_j is denoted by DS(f_j). Then,

$$DS(f_j) = \begin{cases} 1 & \text{if } n(f_j) = 1 \\ 1 - \left(n(f_j) / |O| \right) & \text{otherwise} \end{cases} \quad (1)$$

where $n(f_j)$ is the number of objects in which the feature f_j occurs, O is the set of all the objects, and $|O|$ is the cardinality of O.

3.2 Features organisation and clustering

Based on the DS values of features, features can be classified as common features (I-level), common cluster features (II-level) and special features (III-level). Features for which the DS value is '0' are called common features. It means that the common features appear in every object. Feature for which the DS value is closer to 1 are called special features. The special features appear only in few objects. The other features are called as common cluster features. These features are common to a group of objects.

Table 2 depicts the organisation of features by considering four objects: o_1, o_2, o_3, and o_4. The I-level contains the common features of all four objects. Two clusters are depicted in II-level. One cluster constitutes of o_1 and o_2, and the other cluster constitutes of o_3 and o_4. The common cluster features of each cluster are depicted in II-level. The special features of each object are depicted in III-level.

Table 2 Three level feature organisation

Common features of all the objects (I-level)		
(II-level) Common cluster features	*Object*	*(III-level) Special features*
Common features for o1 and o2	o1	Special features of o1
	o2	Special features of o2
Common features for o3 and o4	o3	Special features of o3
	o4	Special features of o4

It can be noted that, for any object o_i, its complete set of features $f(o_i)$ is a combination of common features at I-level, common cluster features for the cluster in which o_i is a member at II-level and special features of object o_i at III-level.

We now introduce the term 'reduction factor' (rf) to measure the performance. The rf denotes the reduction in the redundancy across the objects in the set of similar objects. The performance is good if rf is high.

Let 'F' denote the total number of features for all the objects. Note that all the repetitive instances are included in F. $F(i)$ denote the number of features in 'i'-level and 'L' denotes the number of levels, which is equal to 3. The 'rf' is defined as,

$$rf = 1 - \frac{\sum_{i=1}^{L} F(i)}{F} \quad (2)$$

If rf value is high, it means that there are several common features among the objects.

It can be noted that, in this paper, we have proposed an approach to organise features into three levels. For simple datasets the features can be organised into two levels. However, for complex datasets, more than three levels maybe required to organise the features.

The main issue here is how to identify the common cluster features. For this we follow the following approach. We divide the objects into different clusters. The features common to all the objects are called I-level features. After subtracting common features from each object, the remaining features of the objects which are common to each cluster are termed as II-level features. For each object, the remaining features after subtracting both common features and common cluster features are called as special features. We discuss the proposed clustering algorithm as follows.

3.2.1 Object clustering algorithm

We first explain the process of computing the similarity between the objects. Let o_i be the i^{th} object, $f(o_i)$ be the set of features of object o_i, $CL(j)$ be the j^{th} cluster and $CF(j)$ be the set of all common features in the j^{th} cluster. The similarity between o_i and $CL(j)$ is denoted by $sim(o_i, CL(j))$ and is calculated as follows:

$$sim(o_i, CL(j)) = \left| f(o_i) \bigcap CF(j) \right| \quad (3)$$

The clustering algorithm employs the notion of quality threshold (QT) to group a set of objects into different clusters (Heyer et al., 1999). The pseudocode of the algorithm is given in Table 3.

The input to the clustering algorithm is a set of objects O, similarity threshold (ST) and feature set F (features of all the objects). The process of clustering is as follows. At first we fix the ST value.

1 For each object o_i from the object list O, we will form a cluster, say 'k'. The following steps are repeated until no further objects are added to the k^{th} cluster. Initialise the cluster with object o_i. Initialise $CF(k)$ which is a set of common features of cluster 'k'. Among the remaining objects, find the common features between every other object o_j with $CF(k)$ having number of common features greater than or equal to ST. Next, select that object o_j with which number of common features with $CF(k)$ is the maximum. Insert o_j into the k^{th} cluster. Update $CF(k)$.

2 At the end of the first step, one cluster is formed for each object. The largest cluster which contains the maximum number of common features, is selected as the QT cluster. Delete the objects of this cluster from O. If no cluster is formed in this step, then go to (3), else go to (1).

3 Objects in O that do not belong in any clusters will be shown as singleton clusters.

Table 3 Special feature extraction algorithm

Input: n: is a number of objects; O: set of 'n' objects;	
F: set of features for all 'n' objects; ST: similarity threshold;	
MCF: minimum number of common features.	
Output: Cluster of objects and singleton clusters.	
1	**Formation of Clusters**
2	Notations used
2.1	nc: number of clusters; i, j: integers;
2.2	$CL[i]$: the i^{th} cluster where ($i <= n$);
2.3	$CF[i]$: set of features of i^{th} cluster;
2.4	$f(o_i)$: set of features for object o_i.
3.1	**if** (| O | ≤ 1) then output O;
3.2	**else do**
3.3	/* Base Case */
	for each $o_i \in O$
	set flag = TRUE;
3.4	$CL[i] = p_i$ /* $CL[i]$ is the cluster started by o_i */
3.5	**while** ((flag == TRUE) and ($CL[i] \neq O$))
	if $CF[i] \cap f(o_j)$ is maximum
	then find $j \in (O\text{-}CL[i])$
3.6	**if** $CF[i] \cap f(o_j) >$ MCF
	then set flag = FALSE;
3.7	**else** set $CL[i] = CL[i] \cup o_j$ /* add j to cluster $CL[i]$ */
	end for
3.8	identify set $C \in CL$, with maximum cardinality.
3.9	output C
3.10	$P = \{P - C\}$
3.11	Repeat from 3.1

The complexity of the proposed algorithm is $O(n^2)$.

Organising the features using three-level method is an iterative process. The value of ST should be chosen to obtain a reasonable number of clusters with high rf.

4 Extending special features to e-commerce environment

In this section, we explain how the proposed approach can be extended to improve the product selection process in e-commerce environment.

4.1 Background

Due to the explosive growth of World Wide Web and popularity of e-commerce, the issue of product selection is becoming more complicated. As every company is concentrating on personalisation and mass customisation, many variations of each product are being introduced in the market with little deviation in their features. This has given more choices to the customer, but has also increased the amount of information that a customer must go through before they are able to select the suitable product. The customer usually gets lost in the vast space of product information and cannot find the products he really wants. As all the information looks similar, it generates confusion in customer's mind. As a result, it becomes difficult for the customer to choose the appropriate product from a set of similar products.

It can be observed that every product has some specialty and possesses corresponding special features. The basic idea is as follows: a customer is shown special feature(s) of each product along with the common features. As a result, the customer can quickly browse through all the special features and make the appropriate selection. We extend the special feature extraction algorithm to e-commerce environment by first computing the degree of specialness of product features. On the basis of degree of specialness and the clustering algorithm, we organise the product features into a three-level organisation.

4.2 Special feature extraction framework

The framework to extract special features is shown in Figure 1. The input to the proposed approach is the text documents where each text document contains the features of one product separated by a delimiter. We briefly explain the steps.

1 Preprocessing: preprocessing involves processing of data in order to prepare it for further analysis. There are several steps required to prepare data before applying the proposed algorithm.

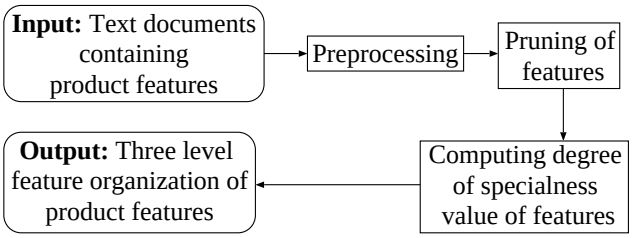
2 Pruning of features: there are some features which are repetitive or subset of another feature. Such duplicates should be removed. For example, the feature '65 w ac power adapter' is a part of the feature 'cq45-106au 65 w ac power adapter'.

3 Computing DS value: the DS values for all the features are computed.

4 Organisation of features: we organise the features using the proposed method.

The final output is the three level organisation of product features.

Figure 1 Extracting special product information



4.3 Experiments

We have conducted the experiments on three real world datasets related to Nokia mobile phones, Sony cameras and HP laptops for the validation of the proposed framework. The results indicate that the proposed approach significantly reduces the number of features a customer needs to be browse for selecting an appropriate product from set of similar products.

- Camera dataset (Sony India, 2011): it contains the details of 7 Sony camera models: DSLR-A350K, DSLR-A350, DSLR-A350X, DSLR-A700H, DSLR-A700K, DSLR-A700. Total number of features comes to 600.
- Mobile phones dataset (Mobile Phones, 2011): it contains the details of 16 Nokia mobile phone models: N-70, N-72, N-73, N-75, N-77, N-80, N-81, N-82, N-85, N-90, N-91, N-92, N-93, N-95, N-96, N-97. Total number of features comes to 382.
- Laptop dataset (HP Official Store, 2011): it contains the details of 10 HP laptop models: CQ50Z, HDX, dv2700t, dv2800t, dv5t, dv7z, dv6700t, dv9700t, tx2000z and tx2500z. Total number of features comes to 320.

Table 4 shows the organisation of features related to Cameras. It can be observed that the camera models in the dataset has many common features that the customer has to go through only once, and not everytime for every model. In Table 5, we provide the reduction factor in the Camera features along with the Mobile phone features and Laptop features, where $|F|$ is total features in the dataset, $\sum_{i=1}^{L} F(i)$ is the total number of features the customer needs to go through, and 'rf' is the reduction factor. The results show that it is possible to reduce the effort made by customers to select the product from the set of similar products. In addition, several products can be compared in an easier manner. Overall, it can be concluded that the proposed approach has a potential to improve the performance of product selection in e-commerce environment.

Table 4 Organisation of features using three-level approach for camera dataset

Common features (I-level)		
Shoulder strap with eyepiece cap and remote commander clip, battery type np-fm500h lithium-ion rechargeable battery, supplied software image data converter sr ver. 2.0, self timer yes (10 seconds 2 seconds off), multi-pattern measuring 40 segment, limited warranty term 1year parts labour, red-eye reduction on/off (all modes), focus auto focus TTL phase detection, OS must be installed at the factory, lens type interchangeable a-mount, service and warranty information, aperture/shutter priority manual, weights and measurements, operating system compatibility, software/USB driver cd-rom, supports USB 2.0 hi-speed, histogram display yes rgb, ver. 2.1.02 (windows only), AF illuminator light yes, XP home and professional, viewfinder optical TTL, picture motion browser, battery capacity 7.2 v, infinity 95% coverage, accessories supplied, primary colour filter, convenience features, operating conditions, ev compensation ev, centre weighted, optics/lens, output(s) video yes, NTSC/PAL selectable, lightbox sr ver. 1.0, auto focus mode yes, inputs and outputs, movie mode(s) n/a, programme shift yes, clear raw nr n/a, USB port(s) yes, lens and media, microphone n/a, w/ev indicator, on/off select, playback only, command dial, fill-flash, rear flash, processor, body cap, hardware, general power, bulb		
Common cluster features (II-level)	*Product*	*Special features (III-level)*
Dimensions (approx.) (whd) (130.8 × 98.5 × 74.7 mm), weight (approx.) 1lb 4.5 oz (582 g) body not including battery, no memory stick media nor adaptors, burst mode max 2.5 fps with viewfinder 2 fps in live view mode 1,600 mah cipa standard, approx 730 pictures, card is full, shooting capacity jpeg, (l size fine) until memory raw jpeg 3 frames, raw 4 frames Megapixel 14.2 mp live view yes, 1/3 ev steps, multi-point 9 area, 8 line, 1 cross sensor, OS × (v 10.3 or later), shutter speed 1/4,000 to 30 sec, wireless off-camera flash, BC-VM 10 battery charger	DSLR-A350.txt	-
	DSLR-A350K.txt	200,000 professionalme, sal-1870 dt 18–70 mm f3.5–5.6 standard zoom lens
	DSLR-A350X.txt	sal-1870 dt 18–70 mm f3.5 zoom lens (27 105 35 mm eq, 55–200 mm f4–5.6 telephoto zoom lens
Dimensions (approx.) (whd) (141.7 × 104.8 × 79.7 mm), 0.3 ev 0.5 ev steps selectable, multi-point 11 area, 5 centre twin-cross lines, flash mode(s) manual pop-up auto weight (approx.) 1 lb 8 oz (690 g) body not including battery burst mode selectable, raw 17, hi (5 fps) lo (3 fps), jpeg extra fine 8, adjustable spot AF selectable jpeg standard/fine unlimited to capacity of media, LCD 3.0 TFT xtra fine (921 k pixels) LCD, wireless off-camera flash (with flash hvl-f56am f36am), visual focus confirm direct via spherical acute matte screen, wireless remote commander (rmt-dslr1), video cable, USB cable	DSLR-A700.txt	Colour mode(s) standard vivid neutral adobe rgb clear deep light portrait landscape sunset night autumn b/w sepia
	DSLR-A700H.txt	Craw (compressed) 24 c raw + jpeg 12 dt 18–200 mm f/3.5–6.3 high magnification zoom lens
	DSLR-A700K.txt	DRO modes include off, standard auto advanced manual dt 18–70 mm f3.5 zoom lens (27 105 35 mm eq) (sal-1870)
	SAL-16105.txt	Memory stick pro media compatibility tested-dt 16–105 mm f3.5 zoom to support up to 16 Gb media capacity lens (24 mm to 157.5 mm eq)

Note: Features are comma (,) separated.

Table 5 Reduction factors for different datasets

Dataset	$\lvert F \rvert$	$\sum_{i=1}^{L} F(i)$	*rf*
Camera dataset	600	122	0.81
Mobile dataset	382	149	0.60
Laptop dataset	320	94	0.71

5 Extending special features from resumes

In this section, we explain how the proposed approach can be extended to improve the process of resume selection.

5.1 Background

In the current scenario, large numbers of resumes are received on-line, through e-mails or through services provided by companies like Info Edge Limited (2011). These large number of resumes are reduced to few hundred potential ones based on some filtering techniques or search services. The set of resumes hence obtained are similar to each other as they satisfy the search criteria or requirements for a company. Thus it becomes necessary to manually analyse each resume to select appropriate resumes. We define this problem as 'Problem of resume selection from set of similar resumes'.

A resume is a multi-topic document where each section describes a different aspect of an individual. It is a semi-structured document with hierarchical organisation of text but the order of sections and the organisation within each section may differ across resumes.

Table 6 shows the sample student resume that contains multiple sections like education, experience, skills and achievements. Each section contains words and sentences as features. The numbering in each section denotes a feature separated by a delimiter ('newline' in our case). Figure 2 shows the corresponding hierarchical structure for Table 6. The top layer, termed as 'Layer 0', contains

Resume Identifier. It can be observed that sections like education, experience, skills and achievements form the first layer of the resume. Each section is described by the text containing words and sentences which forms the second layer of the resume. Based on the structure of the content, the text of each section in the second layer can be organised into several layers.

Table 6 Sample resume with corresponding sections and their respective features

	Education
1	BTech (Computer Science and Engineering) IIIT, Hyderabad (expected May, 2009) 6.66/10 cgpa.
2	Senior Secondary Instrumental School, Kota (CBSE Board 2004) 72%.
3	Secondary St. Sr. Sec. School, Ajmer (CBSE Board 2002) 83%.
	Skills
1	Programming languages: C, C++.
2	Operating systems: Windows 98/2000/XP, GNU/Linux.
3	Scripting languages: Shell, Python.
4	Web technologies: HTML, CGI, PHP.
5	Software tools: Microsoft Office, Latex, GNU/GCC, Visual Studio 2005/2008.
6	Database technologies: MySql.
	Experience
1	Title: Audio-video conferencing over IP networks: description: the objective of this project ...
2	Title: Windows firewall description: packets from or to a network ...
3	Title: Document request form automation description: project developed for IIIT ...
4	Title: Implementation of outer loop join description: implementation of this operation ...
5	Title: Myshell description: emulated a shell environment,...
	Achievements
1	Secured 1,573 air in AIEEE, 2005.
2	Secured 2,216 air in IIT-JEE screening examination, 2005.
3	Cleared NTSE level 1in 2002.
4	Was among the finalists of Rajasthan State Science Talent Search.

In the context of resume processing, we have extended the notion of special features as there may exist special information in some resumes as compared to others. For example, a resume may contain specialty in education, specialty in experience, specialty in skills or specialty in achievements and so on. Special information may exist in one or more sections of a resume. We assume that identifying such special information and organising them efficiently can enhance the performance of the resume selection process.

Resume contains different sections. For extracting special information from resume, we have to extract special information from each section and combine the same in an appropriate manner. In this paper, we have only extended special feature extraction approach to skill section of the resume to extract special skills. Extending the proposed approach for extracting special features to other sections of the resume will be investigated separately (Sainani and Reddy, 2011).

It can be noted that the proposed approach is extended assuming the resumes are semi-structured documents. That is, a certain degree of structure is imposed on resumes.

5.2 Identifying special skills

Table 7 shows the example of features for skill section. It can be observed that the skill section information contains enumerated sequence of text pieces. Each text piece consists of a skill type and its skill values. For example, 'programming languages' is a skill type and 'C, C++, Java' are skill values. The area surrounded by dots in Figure 3 shows the corresponding hierarchical structure for Table 7. If we apply the notion of special features on skills section directly, the comparison between the features would not be effective. This is because there is a two layer organisation in the skill information as shown in Figure 3. It can be observed that the skill information itself forms a hierarchy where skill types form one layer and skill values form another layer. We exploit this inherent organisation in the skills information for effective information extraction.

Table 7 Sample features for skill tag

	Skill type features	*Skill values features*
1	Programming languages	C, C++
2	Operating systems	Windows 98/2000/XP, GNU/Linux
3	Scripting languages	Python, shell
4	Web technologies	HTML, CGI, PHP
5	Database technologies	MySql
6	Others	MS Office, Visual Studio, Latex

Figure 2 Hierarchical structure of skills related information

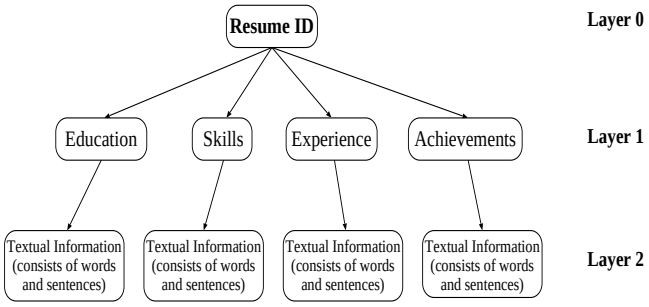
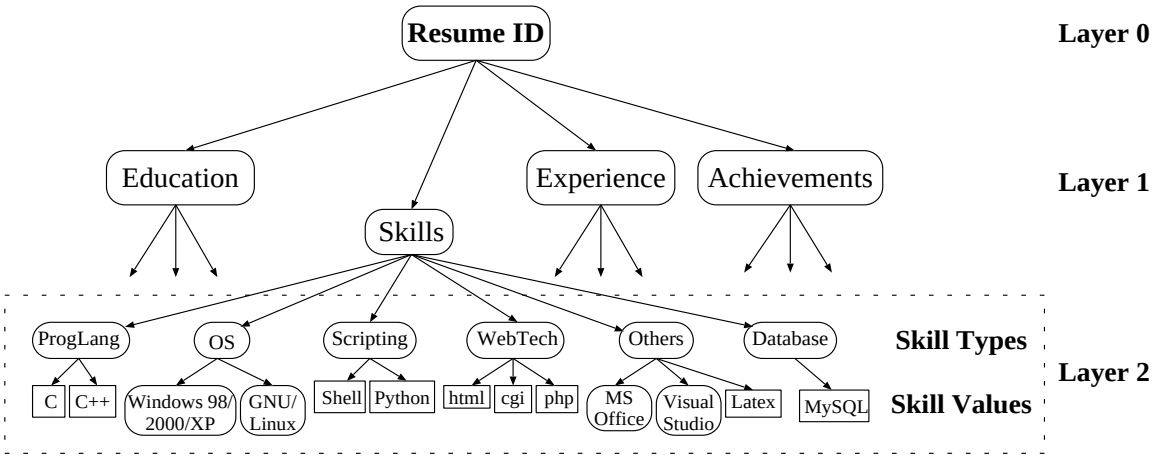


Figure 3 Hierarchical structure of skills related information



We divide the skill information as two kinds: 'skill type' and 'skill values'. We gather all skill type information from each resume and extract special skill type information. Similarly, for each skill type there are several skill values. So, by considering skill values of each skill type, it is possible to extract special skills.

Table 8 Algorithm to calculate STFS and SVFS

Input: R: Set of 'n' resumes; F: set of features for all 'n' resumes;

S: dictionary for all the skill types and

$| S |$ is number of distinct skill types.

Output: STFS and SVFS

1 Notations used

i, j: integers;

S_{r_i} : skills information for resume r_i,

$STFS_i$: array for skill types for resume r_i,

where each tuple contain < r_i, *Skilltype* >

$SVFS_{ij}$: array for skill values for resume r_i and skill type s_j,

where each tuple contain < r_i, s_j, *Skillvalue* >

2 **for** i = 1to n

3 Get the skill section features for resume r_i in S_{r_i}

4 **for** each s_j in S

5 **if** s_j in S_{r_i}

6 store the tuple < r_i, s_j > in $STFS_i$

7 store the tuple < r_i, s_j, *skillvalue* > in $SVFS_{ij}$

8 end

9 end

Overall the proposed approach consists of a number of steps. First we perform pre-processing on the skill information of the resumes. Then we extract the skill type and skill value features. After extracting the skill type and skill value features, we calculate DS values of the features and organise them. One type is skill-type-feature-set (STFS) and another is skill-value-feature-set (SVFSs). Each element in STFS is a two attribute tuple < *ResumeId; Skilltype* > and each element in SVFS is defined as < *ResumeId; Skilltype; Skillvalue* >. Note that, for a given resume, there exists one STFS and several SVFSs. For the same skill type, same skill values may exist, but in different form. Thus, direct comparison cannot be done. So, both STFS and SVFSs are formed after carrying out the preprocessing steps and then applying the described algorithm (refer to Table 8) on the skills information.

The description of algorithm shown in Table 8 is as follows. The input to the algorithm is set R consisting of 'n' resumes, dictionary S that contains all the distinct skill types present in the set R and $| S |$ denotes the cardinality of dictionary S. The output consists of the STFS and SVFS. In STFS each element is a tuple consisting of resume identifier and skill type as its attributes whereas in SVFS each element is a tuple consisting of resume identifier, skill type and skill value. The steps of the algorithm are as follows: We take each resume and repeat the following steps for each resume.

1 identify the skill section of the resume using the 'skills' tag

2 process each line of the skill section to identify the skill type and corresponding skill value

3 the resume id (r_i) and skill type is stored in $STFS_i$ index of the array of STFS where as resume id (r_i), skill type (s_j) and skill value is stored are the index $SVFS_{ij}$ of the array SVFS.

Thus after performing the preprocessing steps and applying the above described algorithm we get STFS and SVFSs. The next task is to calculate the specialness values of all the features in STFS and SVFSs and organise them.

5.3 Overall framework

The input to the proposed approach is the set of resumes stored as text documents where each document contains

different sections along with their descriptions. The steps of proposed framework are discussed below (refer to Figure 4).

1 Identification of features from skills section: we extract STFS and SVFS from skills information for all the resumes.
   - identifying STFS: we identify the skill type features for all the resumes and form an STFS
   - identifying SVFSs: we identify the skill value features for each of the skill type and form SVFSs for all the skill types.

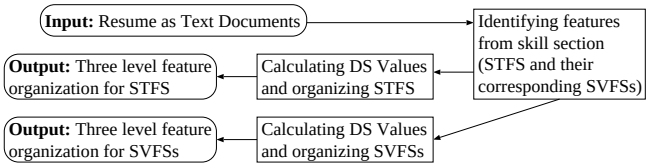
2 Calculating DS value and organising special skill type features: we compute the DS value for skill type features on the basis of DS values we organise the skill type features.
   - computing DS value for skill type features: we compute the DS value for skill type features based on the notion of degree of specialness as defined in equation (1)
   - organisation of skill type features: we organise the skill type features using the special feature extraction and feature organisation approach described in Section 3.2.

   The final output is the three level organisation of STFS.

3 Calculating DS value and organising special skill value features: we compute the DS value for skill value features for each of skill type and on the basis of DS values we organise the skill value features for each of the skill type.
   - Computing DS value for skill value features: we compute the DS value for skill value features for each of skill type based on the notion of degree of specialness defined in equation (1)
   - Organisation of skill value features: we organise the skill value features for each of skill type using the special feature extraction approach described in Section 3.2.

   The final output is the three level organisation of STVSs of each STFS.

Figure 4 Extracting special skill information from resume dataset



5.4 Experiments

Table 9 shows all the distinct skill type features present in the set of resumes in our dataset. Table 10 shows the reduction factor for skill type features. It can be seen that rf value comes to 78%. The results indicate that 78% reduction in the effort could be achieved in resume selection process. The total numbers of skill type features present were 629 and number of features being displayed to a user were only 138. Table 11 shows the organisation of skill type features (set F) using improved special feature extraction approach. The resumes can be classified based on their skill type in one click. Since number of resumes share same special feature we have mentioned them in same row separated by delimiter comma (,) for user convenience as well as to reduce space. There is such a large reduction in the number of features displayed because large number of features are present as common features, so instead of displaying them for each resume, they have been displayed only once. Similarly the cluster features are displayed once for all the resumes present in a cluster instead of separately displaying for each one. Note that in Table 11, we have shrunk some labels for better presentation. For example, we have presented 'mobile platforms' as 'mob plat'.

Table 9 All skill type features

Skill type features			
1	Programming languages	2	Scripting languages
3	Operating systems	4	Web technologies
5	Database systems	6	Libraries/APIs
7	Software tools	8	Compiler tools
9	Mobile platforms	10	Middleware technologies
11	Server side scripting	12	IDE
13	Microsoft tools and services	14	Documentation
15	Java technologies	16	CMS
17	Frameworks and content management systems	18	Object oriented analysis and design
19	Server technologies	20	Version control system
21	Virtualisation tech. and tools	22	Assembly languages
23	Open source tools	24	Open source frameworks

Table 10 Reduction factor values for skill types using proposed approach

Feature type	$\|F\|$	$\sum_{i=1}^{L} F(i)$	*rf*
Skill types	629	138	0.78

Table 11 Organisation of features (skill types) using three-level approach for resume dataset

Common features (I-level)		
Programming languages, operating systems		
Common cluster features (II-level)	*ResumeId*	*Special features (III-level)*
Scrptng lang, web tech lib/APIs, soft tools	R14, R43, R45, R56, R6, R65, R71, R77, R8, R80, R81, R90	DBMS
	R33, R74, R79, R91, R92, R96, R98	DBMS, IDE
	R49	DBMS, mob plat
	R55	DBMS, comp tools
	R18, R28, R41	Comp tools
DBMS, scrptng lang, web tech	R103, R105, R19, R23, R24, R25, R29, R31, R47, R5, R52, R3, R4, R58, R64, R68, R7, R88	None
Scrptng lang, web tech, DBMS, lib/APIs	R106, R11, R60, R82, R93	IDE
	R30, R34	IDE, SERV SIDE SCRPTNG, SOFT TOOLS
	R33, R74, R79, R91, R92, R96, R98	IDE, soft tools
	R63	IDE, serv side scrptng
	R69	IDE, comp tools
	R70, R76, R84, R87, R9	None
	R71	Soft tools
Scrptng lang, comp tools, lib/APIs, web tech, soft tools	R28, R18	None
	R55	DBMS
	R59	IDE
IDE, DBMS, scrptng lang, web tech	R104	Serv side scrptng
	R72	Soft tools, comp tools
	R57	Soft tools, serv side scrptng
	R21	Soft tools, java tech
	R85	None
DBMS, scrptng lang	R32, R89, R94, R99	None
Assmbly lang, soft tools web tech, scrptng lang	R13, R97	None
	R48	IDE
IDE, web tech, soft tools, DBMS	R1	Lib/APIs
	R20	None
Web tech, DBMS, lib/APIs, scrptng lang, mob plat	R49	Soft tools
	R50	Virt tech. and tools
None	R86	Soft tools, DBMS
	R95	Open src frmwrks, comp tools, web tech, scrptng lang, Lib/APIs, DBMS, soft tools
	R53	IDE, soft tools, lib/APIs, DBMS
	R26	Scrptng lang, DBMS, soft tools
	R78	Web tech, IDE, soft tools, frmewrks and content mngmnt sys
	R27	Scrptng lang, IDE, web tech, DBMS serv side scrptng, soft tools, CMS
	R66	Middleware tech, web tech, DBMS, IDE scrptng lang, lib/APIs, mob plat

Table 11 Organisation of features (skill types) using three-level approach for resume dataset (continued)

Common features (I-level)		
Programming languages, operating systems		
Common cluster features (II-level)	*ResumeId*	*Special features (III-level)*
None	R38	Web tech, soft tools, sim tools, assmbly lang
	R73	Scrptng lang, DBMS, OOAD
	R2	IDE, scrptng lang, web tech, serv side scrptng, DBMS, lib/APIs, vers cntrl sys, serv tech
	R61	Assmbly lang, scrptng lang, soft tools, DBMS, doc, web tech, lib/APIs
	R44	Open src tools, soft tools, web tech, DBMS
	R16	Web tech, serv side scrptng, DBMS, soft tools, scrptng lang, assmbly lang, IDE, lib/APIs
	R36	Scrptng lang

Table 12 Reduction factor values for skill values for each skill type using proposed approach

Feature type	$\|F\|$	$\sum_{i=1}^{L} F(i)$	*rf*
Programming languages	289	35	0.88
Database technologies	191	25	0.88
Operating systems	296	68	0.77
Web technologies	348	135	0.62
Scripting languages	184	64	0.67
Software tools	302	208	0.31
Libraries/APIs	123	72	0.42
IDEs	89	52	0.42
Compiler tools	6	4	0.34
Server side scripting	27	15	0.45
Mobile platforms	4	2	0.5
Assembly language	10	5	0.5

The reduction factor in the skill value features for each of the skill type is shown in Table 12. It can be observed that rf values for SVFSs for some skill types is very high, for few skill types low and in some cases medium. The reason for high reduction factor for some skill types is that there are many resume that share common skill values for these skill types and thus the clusters formed are uniform. The reason for low reduction factor for a skill type such as compiler tools or mobile platforms is that the number of features in these sets is very less. Thus there is very little scope of clustering the resumes based on common features. In cases like IDEs or software tools, the variety of skill values across the resumes is very high, hence the low reduction factor. Though in most of the cases reduction factor is 50% or above. For each skill type, its respective skill value features are organised using three-level feature organisation. Table 13 shows the organisation of skill values for skill type 'programming languages'. It can be observed that reduction factor comes to 88% with the proposed approach. Overall, the results indicate that it is possible to reduce the effort of processing resumes by identifying special information, if any, in effective manner with the proposed approach.

Table 13 Organisation of features (skill value :: programming languages) using three-level approach for resume dataset

Common features (I-level)		
C, C++		
Common cluster features (II-level)	*ResumeId*	*Special features (III-level)*
Java	R1, R49, R69	Python
	R101, R39	VB
	R102, R105, R50, R52, R53, R63, R66, R68, R16, R36	None
	R106, R17	Perl
	R33	Symbian C++
	R34, R6, R76	C#
	R37	C#, .net
	R95	j2me
Python	R20, R35, R42, R43, R46, R78	None
	R44	PHP, Perl
	R60	.net, C#, VC++, Python, Java Applets
	R90	MATLAB
Open C++	R12	Prolog
	R80	Symbian
MATLAB	R47, R9	None
VB	R29, R40	None
	r62	Action Script 3.0, MXML
	R86	Perl
	R13	C#
	R65	STL

6 Conclusions

In this paper we have proposed the notion of 'special feature' and an approach to extract the same from a set of text documents. We have extended the proposed special feature extraction approach to improve the process of product selection in e-commerce environment and resume processing. The experiment results show that the proposed approach has the potential to improve the process of decision making in both applications.

As a part of future work, we are planning to carry out extensive experiments by considering different kinds of datasets to develop a robust approach for extracting special features. Also, in case of resume selection problem, we plan to extend the notion of special features to extract special information from other sections of the resume.

References

Aggarwal, C.C. and Yu, P.S. (2001) 'Outlier detection for high dimensional data', *SIGMOD '01: Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, ACM, Vol. 30, No. 2, pp.37–46.

Agyemang, M., Barker, K. and Alhajj, R. (2004) 'Framework for mining web content outliers', in *SAC '04: Proceedings of the 2004 ACM Symposium on Applied Computing*, ACM, pp.590–594.

Agyemang, M., Barker, K. and Alhajj, R. (2005a) 'Mining web content outliers using structure oriented weighting techniques and n-grams', in *SAC '05: Proceedings of the 2005 ACM Symposium on Applied Computing*, ACM, pp.482–487.

Agyemang, M., Barker, K. and Alhajj, R. (2005b) 'Wcond-mine: algorithm for detecting web content outliers from web documents', *Computers and Communications, IEEE Symposium on*, pp.885–890.

Angiulli, F., Fassetti, F. and Palopoli, L. (2009) 'Detecting outlying properties of exceptional objects', *ACM Trans. Database Syst.*, ACM, Vol. 34, No. 0362-5915, pp.1–62.

Barnett, V. and Lewis, T. (Eds.) (1994) *Outliers in Statistical Data*, Wiley Series in Probability & Statistics, John Wiley and Sons.

Breunig, M.M., Kriegel, H-P., Ng, R.T. and Sander, J. (2000) 'LOF: identifying density-based local outliers', *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, ACM, pp.93–104.

Carlson, A., Betteridge, J., Wang, R., Hruschka, E.R., Jr. and Mitchell, T.M. (2010) 'Coupled semi-supervised learning for information extraction', *WSDM '10: Proceedings of the Third ACM International Conference on Web Search and Data Mining*, ACM Press, pp.101–110.

Chau, M. and Chen, H. (2008) 'A machine learning approach to web page filtering using content and structure analysis', *Decision Support Systems*, Elsevier Science Publishers B.V., Vol. 44, No. 2, pp.482–494.

Chu, H-C., Deng, D-J. and Park, J.H. (2011) 'Live data mining concerning social networking forensics based on a Facebook session through aggregation of social data', *Selected Areas in Communications, IEEE Journal on*, August, Vol. 29, No. 7, pp.1368–1376.

Consumer Electronics and IT Products: Sony India (2011) Available at http://www.sony.co.in/ (22 October 2011).

Crestan, E. and Pantel, P. (2010) 'Web-scale knowledge extraction from semi-structured tables', *WWW '10: Proceedings of the 19th international conference on World Wide Web*, ACM, pp.1081–1082.

Heyer, L.J., Kruglyak, S. and Yooseph, S. (1999) 'Exploring expression data: identification and analysis of coexpressed genes', *Genome Research '99,* Cold Spring Harbor Laboratory Press.

HP Official Store (2011) Available at http://www.shopping.hp.com/ (22 October 2011).

Info Edge (India) Limited (2011) Available at http://www.infoedge.in/ (22 October 2011).

Jin, W., Ho, H.H. and Srihari, R.K. (2009) 'OpinionMiner: a novel machine learning system for web opinion mining and extraction', *KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM Press, pp.1195–1204.

Karamatli, E. and Akyokus, S. (2010) 'Resume information extraction with named entity clustering based on relationships', *International Symposium on Innovations in Intelligent Systems and Applications*.

Klosgen, W. (1996) *Explora: A Multipattern and Multistrategy Discovery Assistant*, American Association for Artificial Intelligence, pp.249–271.

Knorr, E.M. and Ng, R.T. (1998) 'Algorithms for mining distance-based outliers in large datasets', *VLDB '98: Proceedings of the 24th International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers Inc., pp.392–403.

Liu, B., Ma, Y. and Yu, P.S. (2001) 'Discovering unexpected information from your competitors' web sites', *KDD '01: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.144–153.

Maheshwari, S. and Reddy, P.K. (2009) 'Discovering special product features to improve the process of product selection in e-commerce environment', *ICEC '09: Proceedings of the 11th International Conference on Electronic Commerce*, pp.47–56.

Maheshwari, S., Sainani, A. and Reddy, P.K. (2010) 'An approach to extract special skills to improve the performance of resume selection', *DNIS '10: Databases in Networked Information Systems, 6th International Workshop*, pp.256–273.

Mobile Phones (2011) Available at http://www.mobile.am/ (accessed on 22 October 2011).

Paravastu, R., Kumar, H. and Pudi, V. (2008) 'Uniqueness mining', *DASFAA*, pp.84–94.

Qiu, T. and Yang, T. (2010) 'Automatic information extraction from e-commerce web sites', *Proceedings of the 2010 International Conference on E-Business and E-Government*, IEEE Computer Society, pp.1399–1402.

Ramaswamy, S., Rastogi, R. and Shim, K. (2000) 'Efficient algorithms for mining outliers from large data sets', *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, ACM, pp.427–438.

Sainani, A. and Reddy, P.K. (2011) 'Extracting special information to improve the efficiency of resume selection process', MS thesis, November.

Sun, J-T., Wang, X., Shen, D., Zeng, H-J. and Chen, Z. (2006) 'CWS: a comparative web search system', *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, ACM Press, pp.467–476.

Wei, L., Qian, W., Zhou, A., Jin, W. and Yu, J.X. (2003) 'Hot: hypergraph-based outlier test for categorical data', in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, Springer-Verlag, pp.399–410.

Wu, F. and Weld, D.S. (2010) 'Open information extraction using Wikipedia', *ACL '10: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp.118–127.

Yu, K., Guan, G. and Zhou, M. (2005) 'Resume information extraction with cascaded hybrid model', *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, pp.499–506.

Zhang, J. and Wang, H. (2006) 'Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance', *Knowl. Inf. Syst.*, Springer-Verlag New York, Inc., pp.333–355.

Zhu, C., Kitagawa, H. and Faloutsos, C. (2005) 'Example-based robust outlier detection in high dimensional datasets', in *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, IEEE Computer Society, pp.829–832.