

Intelligent Web Based Task Completion Using Pattern Recognition Techniques

Chetan Bansal and Adwait Gandhe

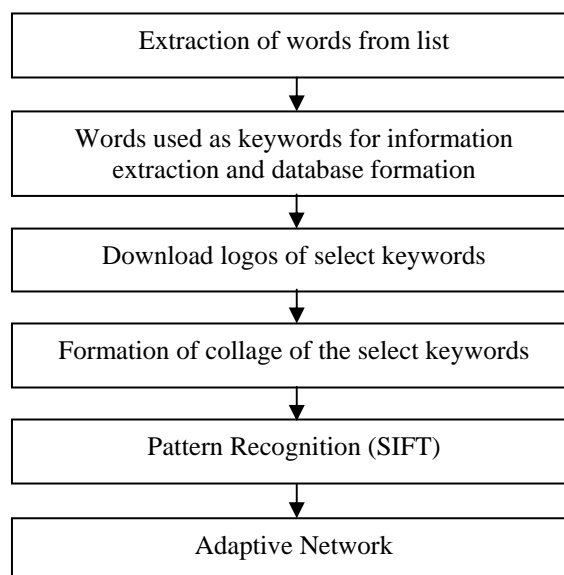
Abstract—The use of electronic devices for ease of operation has increased largely in the world. Various techniques are used for intelligent extraction of information from the surroundings and developing effective interfaces for obtaining the same. This paper proposes a system that rules out this interface altogether and provides the user with intelligent task completion. Pattern recognition techniques have been used for data extraction from the surroundings and cascaded with data mining algorithms that fetch the relevant information from the web. Users commonly make to-do lists on their smart devices. However, it is common for users to forget to refer to them in places where tasks can be completed potentially. The system is so designed to send a message to the user that a particular task can be completed at a particular time. An adaptive learning based algorithm also improves the efficiency of the system with usage.

Index Terms—Pattern recognition, data mining, adaptive learning

I. INTRODUCTION

The development of consumer electronics has seen large number of products being developed for ease of operation and consumer satisfaction. This paper proposes a system that aids the customer for better management of task completion. It is common for a lot of people to make a to-do list on their smart devices. It is also common for them to have been to places where a particular task could have been completed. However, the users forget to refer to the list. Hence, to-do lists today are used by people when they take time off from routine and dedicate time only for task completion. As an example, suppose a user has added a clause of purchasing coffee from the market. The user, while working say visits a coffee shop to have coffee with colleagues. It is potentially possible for the user to buy coffee and complete the task, however he may not remember the same as he is in a different mindset. This work proposes a system that requires very less interaction and helps to complete such tasks. We assume that such to do list has been generated by the user. The algorithm then extracts key words from the list to determine the things associated with the tasks. Using data mining techniques, relevant brand names and brand logos are extracted. We have developed an algorithm that simply extracts this information from the well known website wikipedia. The brands of these commodities are then downloaded. A collage of all the images is created to develop a single image for comparison. In order to extract the availability of commodities with these brand names, we have

used a pattern recognition technique that uses the well known SIFT feature matching. The SIFT features of the collage are extracted and matched against the SIFT features of the live images obtained from the surroundings. In cases where a particular product bearing a brand of interest is detected, the user is sent a notification that involves the information of the particular task he has stored as well as the brand name detected. The user now has two choices, to accept or reject. This response from the user is stored and provided to an adaptive network. For example, a user interested in buying coffee may not be enthusiastic about a particular brand. The system then stores this information and provides more relevant matches. A flowchart depicting the system is shown here:



II. INFORMATION EXTRACTION & DATABASE FORMATION

We need to fetch brands related to the keywords which were extracted out of the to-do list. We related brands with the keywords by extracting information from Wikipedia. Wikipedia is the largest collaboratively edited encyclopedia which is maintained by volunteers from around the world^{[1][2]}. We decided to use Wikipedia because of the vast amount of information and also because it provides a consistent interface for searching through it. Wikipedia provides a huge knowledge base for finding brands and entities owing to its community maintained nature. It also provides us with a source using which we can find further information about individual brands.

To match brands with commodities or keywords that were extracted from the to-do list we developed a framework built on open source technologies using the LAMP stack. PHP was

used as the server side language for data processing, while MySQL was used as the database in the backend for caching. We maintain a cache of all the results for faster response time. We have set an automated timeout for purging and refreshing the cache. To connect to the Wikipedia server from PHP, we make use of the CURL library.^{[3][4]} The steps required to fetch and process the information from Wikipedia are listed below:

- Check the cache, if we have the required information. If yes, display the information from the cache.
- If the required information is not cached, using the curl library we create an interface to the Wikipedia search page, passing the keyword as the search term. We obtain the HTML source code of the Wikipedia results page.
- The source text we obtain from Wikipedia contains a lot of HTML tags and meta information which is stripped out by using a combination of Regular expression and string functions of PHP. Now, we have only the relevant text with us.

III. PERSONALIZATION & INCREASING ACCURACY

Though the final result set we obtain is highly specific and accurate as per our extensive sample data sets. However, the results can be personalized and improved further. From the text that we obtained in the last step, we have to extract only the brand names. This is done using a two tiered process. Each of the search result we obtain from Wikipedia has a title. As per our data set, this title usually contains the brand name. So, we extract the titles of all the search results. The second step requires us to use a free Natural Language Processing framework by Reuters which is called Calais. We pass all the dataset we have into it and it returns RDF formatted results with entities, facts and events tagged within the input dataset.

The resulting datasets obtained from the two tiered method we performed in the previous step are matched against each other and merged into a single dataset so as to obtain accurate results. We also introduced two parameters so as to increase the accuracy and personalization of the results, which are location and number of results which are extracted from Wikipedia. For the location based parameter we use the location of user, which can be taken using the onboard GPS device. The region can be anything, such as state, country or even continent, as per the desired level of localization. We have attached some sample datasets with varying countries.^[5] To enable this feature in the backend, we append the location with the keyword while fetching search results from Wikipedia. Further, Wikipedia allows us to fetch any number of results. By default, we extract the first 20 relevant search results. But, we can define a parameter called accuracy (min: 0, max: 1). By varying this parameter we can expand the dataset which is fetched from Wikipedia. Finally the result set is stored in the cache for future retrieval along with the parameters i.e. accuracy and location.

IV. DOWNLOAD LOGO

Till now we have extracted the keywords or commodities from the to-do list. Further, using Wikipedia we have

obtained a list of brand names with varying level of accuracy and localization. The next step is to find logos of these brands and form a collage out of it for pattern recognition

The fetching of logos is automated by using a server side script which connects to our cache and downloads logos using Google Image search. We select the logos based on the order and the resolution. Choosing a higher resolution logo makes the process of pattern recognition easier.

V. COLLAGE FORMATION

The logos thus obtained are scaled down to a fixed width of 200px and converted to JPEG format to save storage space. Further we create a new canvas and paste the image on the canvas using the GD library. It is a image processing library for Php, using which we can manipulate images of various formats like JPEG, PNG, GIF and so on.

Finally, we save the collage on the hard disk and also create a cache entry containing the keyword, timestamp and the filename for future retrieval.

VI. PATTERN RECOGNITION USING SIFT

We have used an algorithm proposed by David Lowe at ICCV 1999 for pattern recognition.^[6] The algorithm extracts SIFT features, Scale Invariant Feature Transform features from an image in question. These features, as the name suggests are invariant to rotation, scale and partially invariant to occlusions. Hence, it is possible to use this method here as the products in question may have the brand names written in various orientations. The idea is to calculate the SIFT features of the collage image that is generated. This image is then matched with the stream of images from the surroundings. Because of the SIFT algorithm, it is possible to detect a particular logo from the entire collage and determine which of the logos is detected from the neighborhood of the SIFT features that are matched. The SIFT algorithm first extracts the SIFT features from an image and then matches these features from the input images.^{[7][8]}

The implementation of the SIFT algorithm is available in MATLAB as a demo code developed by David Lowe. For the implementation of SIFT in this work we have used functions developed by Rob Hess at University of Oregon and made necessary changes, using OpenCV for pattern recognition.

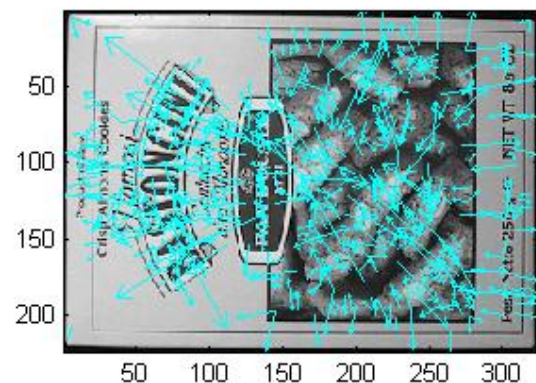


Fig. 1. SIFT features extracted from an image

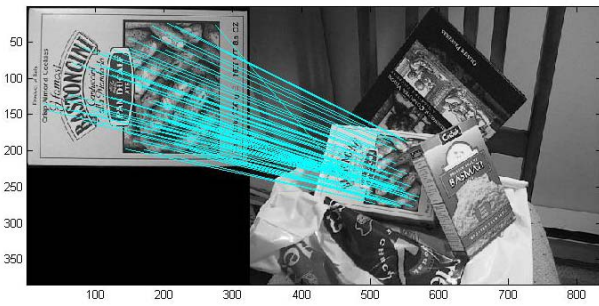


Fig. 2. Extracted SIFT features matched.



Fig. 3. Implementation of the algorithm

VII. LEARNING BASED SYSTEM WITH ADAPTIVE FEEDBACK

The above mentioned method of developing a system for task completion gives fairly accurate results (described in detail further). However, the accuracy of the results increases on making the system adaptive. It may be common for some people to prefer a particular brand. It may also be possible that the backends developed fetch some irrelevant data. The idea is to request a response from the user whenever a notification is sent. The response is stored in a database that then presents the user with notifications more relevant to the user's choice. Though it takes time for the system to adapt to each commodity till the commodity is acted upon at least once, the accuracy increases after the first adaptation. We also propose a system where such a network is built into the database of large retail stores. For example, instead of the system mining relevant data from the internet, the data can be extracted from this database that obviously consists of a clear mention of the category of the product and the available brands. In this manner, the system does not utilize unnecessary computing power for searching a commodity not available at the store.

VIII. CASE STUDY

The system has been developed and tested in local stores. For the purpose of selective results, the device extracts about its location from the GPS or service provider. The searches are then made accordingly. We have presented here the statistics of the comparison of accuracy individually over different geographic locations as well as a comparison. The 100% and the 40% accuracies mentioned are the ones we have distinguished in respect to the number of Wikipedia entries extracted.

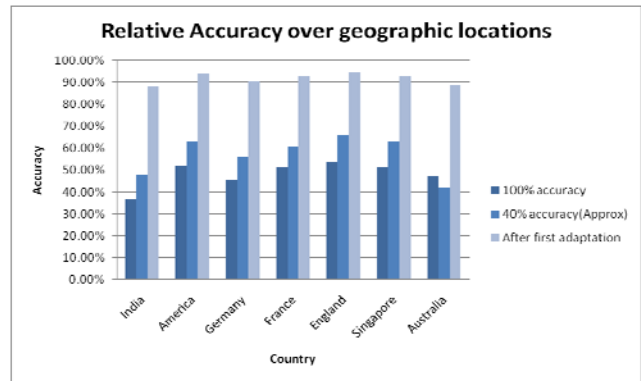


Fig. 4. Relative accuracy of system

The above relative comparison clearly shows that the accuracy of the relevant data increases remarkable after the first adaptation. The idea is that once the device is used for a substantial amount of time the accuracy increases and the data suits the interests of the user.

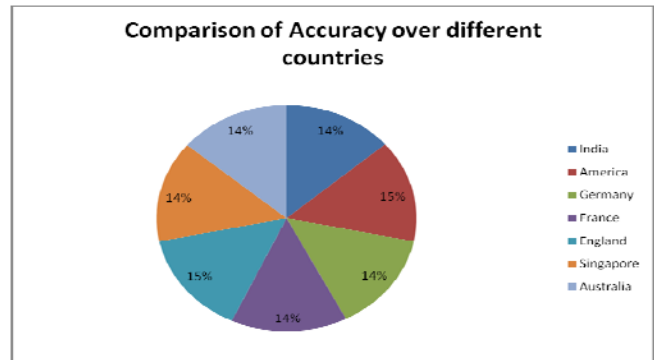


Fig. 5. Comparison of Accuracy

The above comparison shows the accuracy is constant over almost all the countries. Hence the system proposed can be used in varied geographic locations. The invariance to location can be attributed to the intelligent extraction. The adaptive network also results in the system behavior to be modified according to the user requirements.

IX. CONCLUSION

The system mentioned has been implemented with fair accuracy. The results also show that the adaptive network helps in increasing the effectiveness and personalize the product. The increase in the computing power of the available hardware enables the product to be used on an embedded platform. More refined data mining and pattern recognition techniques will further help enhance the utility of the product.

X. FUTURE WORK

Currently a simple application is being developed. It is common practice for a family or friends to help each other complete certain tasks. Hence, we are currently developing an extension that helps synchronize the different to-do lists according to user granted permissions. The idea then is to mention the probable task completion and the person who wishes to complete it.

As a more improved version, the actual task of creating the to-do list can also be automated. For this, currently work is being done for voice recognition and data mining from this vocal information. The idea is to create a database of tasks to be completed by extracting the information from daily conversations and then implement the already developed system.

REFERENCES

- [1] Simone Paolo Ponzetto and Michael Strube EML Research 'Deriving a Large Scale Taxonomy from Wikipedia'
- [2] Buitelaar, P., P. Cimiano and B. Magnini (Eds.). "Ontology Learning from Text: Methods, Evaluation and Applications," Amsterdam, The Netherlands: *IOS Press*, 2005.
- [3] Strube, M. and S. P. Ponzetto, "WikiRelate! Computing semantic relatedness using Wikipedia," in *Proc. of AAAI-06*, pp.1419–1424, 2006.
- [4] Finkelstein, L., E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, "Placing search in context: The concept revisited," *ACM Transactions on Information Systems*, vol. 20(1): pp. 116–131, 2002.
- [5] Placing Search in Context: The Concept Revisited Lev Finkelstein, Evgeniy Gabrilovich¹, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman and Eytan Ruppin Zapper Technologies Inc. 3 Azrieli Center, Tel Aviv 67023, Israel.
- [6] Lowe, D. G. "Object Recognition from Local Scale Invariant Features," *ICCV* 1999.
- [7] Lowe, D. G. "Distinctive Image Features from Scale Invariant Keypoints," *IJCV* 2004.
- [8] Lienhart, R., Liang, L., and Kuranov, A., "A Detector Tree of Boosted Classifiers for Real Time Object Detection and Tracking," *ICME* 2003.