Volume 5, Issue 12, December 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Accurate Career Trends Extraction for Information Professionals using Agile Text Mining

Rajeev Tripathi, Santosh Kumar Dwivedi
Computer Science & SRMGPC,
India

Abstract -Through recognizing the significance of a qualified workforce, skills, career, research has become one of the focal points in education, economics, and placements. In this work we concentrate on skills needs, nature of job, enticing career are dynamic variables dependent on many aspects such as geography time, a vocation or aspiration. The purpose of this paper will identify current trends and issues in research focusing on career and technical education. The term career and technical education (CTE) was viewed from a broad perception that included workforce education, technical education, technical college and community college etc. Results should allow researchers, practitioners and policy makers to identify instant and emerging research needs in career and technical education. Queries were constructed based on 546 students' data summaries available as training data (i.e. resume). Performance was measured on a test dataset of various filled documents (questioners).

Keywords—Agile text mining, Data Mining, Annotation

I. INTRODUCTION AND BACKGROUND

This Career and technical education is an integral part of secondary and postsecondary public education and is designed to educate about, through, and for careers. Career technical education provides students and adults with the technical skills, knowledge and preparation necessary to succeed in specific occupations and careers. It also prepares students for the world of work by introducing them to place of work competencies that are essential no matter what career they choose. And, career technical education takes academic content and makes it accessible to students by providing it in a hands-on context. Text mining is the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources. Text mining is a variation on a field called data mining that tries to find interesting patterns from large databases. Text mining, also known as intellectual Text Analysis Text Data Mining or Knowledge Discovery in Text (KDT), refers generally to the process of extracting interesting and non-trivial information and knowledge from unstructured text.

The problem of Knowledge Discovery from Text (KDT) is to extract explicit and implicit concepts and semantic relations between concepts using Natural Language Processing (NLP) techniques.

Background

Agile text mining can be applied without building an annotated preparation corpus, so is well-suited to novel or one-off extraction tasks.

It is an alternative to traditional project management, typically used in software development. It helps teams respond to unpredictability through incremental, iterative work cadences, known as sprints. Agile methodologies are an alternative to waterfall, or traditional sequential development.

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

II. SCRUM

Scrum is the most popular way of introducing Agility due to its simplicity and flexibility. Because of this popularity, many organizations claim to be "doing Scrum" but aren't doing anything close to Scrum's actual definition. Scrum emphasizes empirical feedback; team self management, and striving to build properly tested product increments within short iterations. Doing Scrum as it's actually defined usually comes into conflict with existing habits at established non-Agile organizations.

Scrum has only three roles: Product Owner, Team, and Scrum Master. These are described in detail by the Scrum Training Series. The responsibilities of the traditional project manager role are split up among these three Scrum roles.

Its aim is to get insights into large quantities of text data. KDT, while deeply rooted in NLP, draws on methods from statistics, machine learning, reasoning, information extraction, knowledge management, and others for its discovery process. KDT plays an increasingly significant role in emerging applications, such as Text Understanding.

Burn-downs charts

Burn-downs charts are among the most common sprint tracking mechanisms used by agile practitioners. Though their application and usage varies (some plot a burn-down chart using story points, whereas others use task count), plotting burn-down using effort remaining is the most effective and efficient way of using burn-down charts. This article looks at creating and updating a burn-down chart using the effort-remaining approach, interpreting burn-down under different scenarios, and examining common mistakes to avoid while using burn-downs. We conclude by looking at some of the benefits of using this innovative tool.

Creation of burn-down chart

The first step is to have a task breakdown in place. This is generally done during the sprint planning meeting. Each task should have associated hours (ideally not more than 12, roughly two days' work at six per day), which the team decides on during the planning meeting.

Once the task breakdown is in place, the ideal burn-down chart is plotted. The ideal reflects progress assuming that all tasks will be completed within the sprint at a uniform rate (refer to the red line in Figure below).

Many Agile tools (Rally, RTC, Version One, ingle, etc.) have built-in capability for burn-down charts. However, in its simplest form, a burn-down chart can be maintained in a spreadsheet. Dates in the sprint are plotted on the X axis, while remaining efforts are plotted on the Y axis.

Refer to the example below:

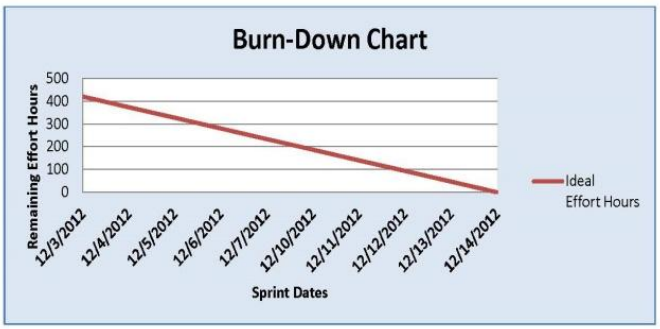
Sprint Duration *– 2 weeks*

Team Size *- 7*

Hours/Day *– 6*

Total Capacity *– 420 hours*

On Day 1 of the sprint, once the task breakdown is in place, the ideal burn-down will be plotted as below:



The Y axis depicts total hours in the sprint (420 hours), which should be completed by the end of the sprint. Ideal progress is shown in the red line, which assumes all tasks will be completed by the end of sprint.

III. METHODOLOGY

Most of the related work performed in this field was directed towards performing a precise analysis of survey or observations. The main source of data for each article was coded as being generated by survey (questionnaire), interview (face-to-face, email, telephone) documents (internal memos, newsletters, resume).

Multiple issues had to be taken care of in this project. Different queries captured parts of the text corresponding to the same slot. For example, a query aimed at capturing a particular linguistic construct may extract frequency as “students of different courses or stream,” while another query may capture its substring “technical students.” In this case, the former extraction, which is the best suitable answer, received priority as per the challenge specifications. Another important problem encountered was that of multiple matches for the same career option. For example, IT Manager and Information Technology Administrator as well as database administrator were identified as separate job option during the indexing process. However, “IT Manager” is considered as a single job name as per the challenge specifications.

IV. DATA COLLECTION

We automatically collected a set of CVs of students and technocrats which are publicly available online. This data set was created by firstly querying Google using the Google API2 for word documents containing either the terms ”CV”, ”resume” or ”curriculum vitae” as well as the terms ”developer”, ”programmer” or ”software” but excluding documents containing the word ”template” or ”sample”.

Annotation Phases

We employed 4 annotators with various degrees of experience in annotation and computer science and therefore familiar with software engineering, management skills and terminology. The lead researcher of the project, the first author of this paper, managed the annotators and organized regular meetings with them. We followed the agile corpus creation approach and carried out cycles of annotations, starting with a simple paper-based pilot annotation. This first annotation of 20 documents enabled us to get a first impression of the type of information contained in CVs of software engineers and programmers as well as the type of information we wanted to capture in the manual and automatic

annotation. We drew up a first set of potential types of zones that occur within CVs and the types of Named Entities that can be found within each zone (e.g. an EDUCATION zone containing NEs of type PER, DOB, QUAL, JOB, LOC, ORG and QUAL).

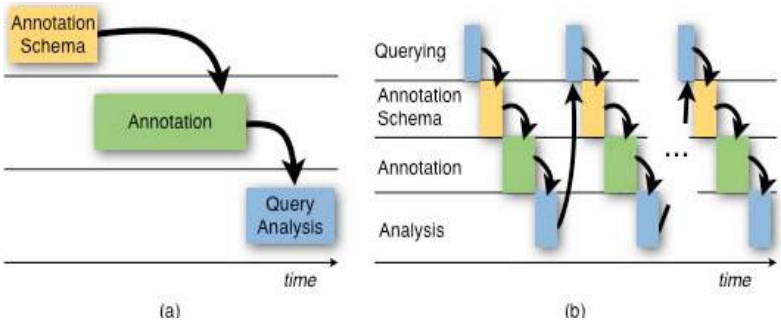


Figure 2: The phases of traditional corpus creation (a) and the cyclic approach in agile corpus creation (b).Reproduction of figure 2 in Voormann and Gut (2008).

The technology used was based on TM components that were originally developed for the biomedical domain during its predecessor project (Alex et al., 2008b). In TXV we adapted the tools to the recruitment domain in a short time frame. The aim was to extract key information from curricula vitae (CVs) for matching applicants to job adverts and to each other. The TM output is visualized in a web application with search navigation that captures relationships between candidates, their skills and organizations etc. This web interface allows recruiters to find hidden information in large volumes of unstructured text. We automatically downloaded the Word documents returned by this query, resulting in a pool of 1,000 candidate CVs available for annotation. We split these documents randomly into a TRAIN, a DEVTEST and a TEST set in a ratio of approximately 64:16:20. We used the annotated TRAIN data for training ML-based models and deriving rules and the DEVTEST data for system development and optimization.

We set aside the blind TEST set for evaluating the final performance of our named entity recognition (NER) and relation extraction (RE) section of the CV data set. The final manually annotated data set contains 403 files, of which 352 are singly and 51 doubly annotated, resulting in an overall total of 454 annotations. This does not include the files used during the pilot annotation. The doubly annotated CVs were used to determine inter-annotator agreement (IAA) in regular. Some of the documents in the pool were not genuine CVs but either job adverts or CV writing advice.

We let the annotators carry out the filtering process of only choosing genuine CVs of software developers and programmers for annotation and reject but record any documents that did not fit this category.

The annotators rejected 99 files as being either not CVs at all (49) or being out-of-domain CVs from other types of professionals (50). Therefore, just over 50% of the documents in the pool were used up during the annotation process.

V. DOCUMENT PREPARATION

Before annotation, all candidate CVs were then automatically converted from Word DOC format to Open Office ODT as well as to Acrobat PDF format in a batch process using Open Office macros. The resulting contents.xml files for each ODT version of the documents contain the textual information of the original CVs. An XSLT style sheet was used to simplify this format to a simpler in-house XML format, as the input into our pre-processing pipeline. We retained all formatting and style information in span elements for potential later use.

The pre-processing includes tokenization, sentence boundary detection, part-of-speech tagging, lemmatization, chunking, abbreviation detection and rule-based NER for person, location names and dates. This information extraction system is a modular pipeline built around the LT-XML24 and LTTTT25 toolsets. The NER output is stored as stand-off annotations in the XML. These pre-processed files were used as the basis for annotation.

Annotation Tool

For annotating the text of the CVs we chose MMAX2, the Java-based open source tool (M¨uller and Strube, 2006).6 MMAX2 supports multiple levels of annotation by way of stand-off annotation. As a result MMAX2 creates one separate file for each level of annotation for each given base data file. Only the annotation level files get edited during the annotation phase. The base data files which contain the textual information of the documents do not change. In our project, we were interested in three levels of annotation, one for named entities (NEs), one for zones and one for relations between NEs.

The MMAX2 GUI allows annotators to mark up nested structures as well as intra- and inter-sentential relations. Both of these functionalities were crucial to our annotation effort.

Annotation Scheme

In this section, we provide a summary of the final annotation scheme as an overview of all the mark able present in the annotated data set.

Named Entities

In general, we asked the annotators to mark up every mention of all NE types throughout the entire CV, even if they did not refer to the CV owner. With some exceptions (DATE in DATERANGE and LOC or ORG in ADDRESS),

annotators were asked to avoid nested NEs and aim for a flat annotation. Discontinuous NEs in coordinated structures had to be marked as such, i.e. the NE should only contain strings that refer to it. Finally, abbreviations and their definitions had to be annotated as two separate NEs. The NE types in the final annotation guidelines are listed. While carrying out the NE annotation, the annotators were also asked to set the NE attribute of type CANDIDATE (by default set to true) to false if a certain NE was not an attribute of the CV owner (e.g. the ADDRESS of a referee).

VI. CONCLUSIONS

Extracting information through interactive design of quires can achieve highly precise results in a short amount of time. Mush of time in this work was spent on pre-processing documents to allow the results to confirm specified format. The actual trends shows that most jobs set belong to their related skill set like mostly software engineer job preferred by computer science courses. But some distinct jobs set also available like some students who have computer science knowledge also like marketing job set. In future work we should also work on distinct data sets and more precise result oriented approach. .

ACKNOWLEDGMENT

This research paper is made possible through the help and support from everyone, including: parents, teachers, family, friends, and in essence, all sentient beings. Especially, please allow me to dedicate my acknowledgment of gratitude toward the following significant advisors and contributors: First and foremost, I would like to thank Mr. Vikrant Bhateja for his most support and encouragement. He kindly read my paper and offered invaluable detailed advices on grammar, organization, and the theme of the paper. Second, I would like to thank Dr. Vonodini Katiyar and Dr. Brijendra Singh to read my paper and to provide valuable advices. Finally, I sincerely thank to my parents, family, and friends, who provide the advice and financial support. The product of this research paper would not be possible without all of them.

REFERENCES

[1] Beatrice Alex, Malvina Nissim, and Claire Grover. 2006.The impact of annotation on the performance of protein tagging in biomedical text. In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy.

[2] EGFSN. Tomorrow's Skills. Towards a National Skills Strategy; Expert Group on Future Skills Needs: Dublin, Ireland, 2007. Bea Alex, Claire Grover, Barry Haddow, Mijail Kabadjov, Ewan Klein, Michael Matthews, Richard Tobin, and Xinglong Wang. 2008a. The ITI TXM corpora: Tissue expressions and protein-protein interactions. In Proceedings of the Workshop on Building and Evaluating Resources for Biomedical Text Mining at LREC 2008, Marrakech, Morocco.

[3] Litecky, C.; Aken, A.; Ahmad, A.; Nelson, H.J. Mining for Computing Jobs. IEEE Softw. 2010, 27, 78–85. [3]Beatrice Alex, Claire Grover, Barry Haddow, Mijail Kabadjov, Ewan Klein, Michael Matthews, Richard Tobin, and Xinglong Wang. 2008b. Automating curation using a natural language processing pipeline. Genome Biology, 9(Suppl 2):S10.

[4] Sue Atkins, Jeremy Clear, and Nicholas Ostler. 1992. Corpus design criteria. Literary and Linguistic Computing, 7(1):1–16.

[5] Zhang, S.; Li, H.; Zhang, S. Job Opportunity Finding by Text Classification. Procedia Eng. 2012,29, 1528–1532.

[6] Douglas Biber. 1993. Representativeness in corpus design. Literary and Linguistic Computing, 8(4):243–257.

[7] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20:37–46.

[8] Cedefop. User Guide to Developing an Employer Survey on Skill Needs; Publications Office of the European Union: Luxembourg, 2013.

[9] Haralampos Karanikas and Babis Theodoulidis Manchester, (2001), "Knowledge Discovery in Text and Text Mining Software", Centre for Research in Information Management, UK

[10] Brin S., and Page L.(1998), " The anatomy of a largescale hyper textual Web search engine", Computer Networks and ISDN Systems, 30(1-7): 107-117.

[11] Shantanu Godbole, and Shourya Roy, India (2008), "Text to Intelligence: Building and Deploying a Text Mining Solution in the Services Industry for Customer Satisfaction Analysis", IEEE, 441-448.

[12] Zhang, S.; Li, H.; Zhang, S. Job Opportunity Finding by Text Classification. Procedia Eng. 2012,29, 1528–1532.

[13] Hutchinson, S. R., & Lovell, C. D. (2004). A review of methodological characteristics of research published in key journals in higher education: Implications for graduate research training. Research in Higher Education, 45, 383-403.

