SemChat: Extracting Personal Information from Chat Conversations

Keith Cortis, Charlie Abela

Faculty of Information and Communication Technology,
Department of Intelligent Computer Systems,
University of Malta
kcor0003@um.edu.mt, charlie.abela@um.edu.mt

Abstract. The Semantic Desktop builds over Bush's Memex vision and focuses on enhancing the personal information management (PIM) process through the integration and presentation of content found on the user's desktop. In line with the Semantic Desktop's philosophy we present SemChat, which is a semantic chat client component. We discuss how SemChat allows personal information related to persons, locations, organisations, dates and events to be extracted from chat conversations and to be integrated into the user's Personal Information Model (PIMO), with annotated events being directly exported to an event scheduler. We also discuss SemChat's search facility, which allows users to search for relevant concepts within their personal chat-information space. Furthermore we elaborate on our initial evaluation efforts which proved to be very promising.

Keywords: personal information management, social semantic desktop, personal information model, semantic chat

1 Introduction

The internet has brought about a radical change in the way people interact. Online communities have flourished, first fueled by electronic mail (e-mail), and nowadays complemented by instant messaging (IM). The advent of e-mail triggered a chain reaction that naturally resulted in the development of IM in 1993, since the former is not as immediate. For this reason IM has become very popular over recent years. Common acquaintances can communicate with each other, in real time using IM whereby messages are transferred from one user to another in a seemingly peer-to-peer manner.

However with the increase in applications that allowed these virtual online communities to flourish, came also an increase in the fragmentation of personal information. It is left up to the user to integrate and manage this disparity in personal information scraps, such that these are not forgotten or lost. In this regards, numerous tools have been developed to aid users in the management of their personal information space.

The vision behind the Semantic Desktop (SD) is precisely that of tackling the difficulties when managing personal information. It builds over Bush's Memex[1] vision and focuses on enhancing the personal information management (PIM) process through the integration and presentation of content found on the user's desktop, by using Semantic Web standards and technologies. This vision is further extended within the Social Semantic Desktop (SSD) which projects the SD into the social dimension and augments SD with facilities for information distribution and collaboration [12].

In line with the Social Semantic Desktop's philosophy our research aims at exploiting and extending NEPOMUK[2], a Social Semantic Desktop framework, with SemChat, a semantic chat client component. The main objectives behind SemChat include the following:

- compatibility with different chat clients
- provide for the extraction and annotation of the user-relevant concepts from a chat conversation which have not already been stored within the users Personal Information Model (PIMO)[3]
- provide for the identification and extraction of any events mentioned during a chat conversation, together with the option to annotate such events within an available task/event scheduler.
- provide for the persistence of any concepts that were not readily annotated by the user, for reference in future SemChat sessions
- provide for a search facility over the chat-related concepts (and events)

The rest of the paper is organized as follows. In Section 2 we highlight the main ideas behind SemChat's architecture and implementation, whilst in Section 3 we present and discuss the results obtained after an initial evaluation session. We go over some related research in Section 4 and provide some future aspirations and concluding comments in Section 5 and Section 6 respectively.

2 SemChat

In Figure 1 we present a general architecture of SemChat and its main components. The motivation behind this architecture partly came from work performed on Semanta [10] and SemNotes [3] which are applications that also exploit the ideas behind SD and SSD. The former is a semantic email component while the latter is a note-taking tool, and both integrate closely with NEPOMUK.

NEPOMUK's environment allows the user to manage all the data found on her desktop and to link the documents within the PIMO [8]. This ties perfectly with one of our main objectives within SemChat precisely that of extracting user-relevant concepts and events from chat conversations and to expose and link, this extracted knowledge, with that found on the user's desktop. In this manner, the

[1] http://cyberartsweb.org/cpace/ht/jhup/memex.html
[2] http://nepomuk.semanticdesktop.org/
[3] http://dev.nepomuk.semanticdesktop.org/wiki/PimoOntology

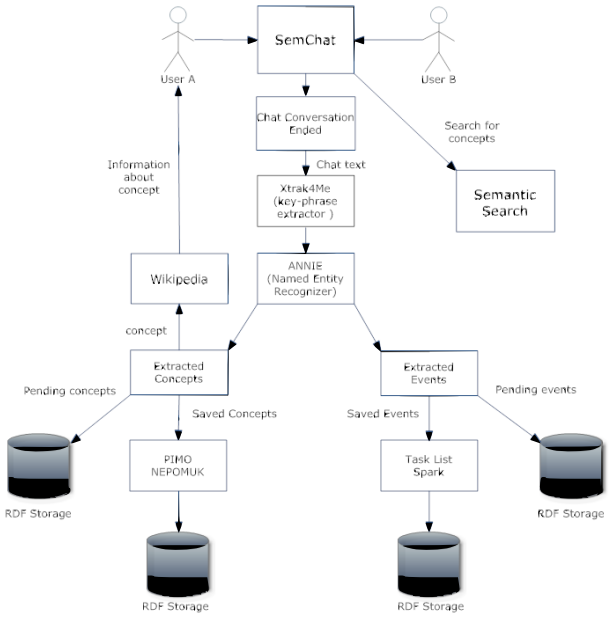

Fig. 1. General Architecture of SemChat and its main components

user's PIMO is augmented with newly found concepts mentioned during chat sessions while at the same time during conversations the user can versatility exploit existing concepts found within this same PIMO. Therefore, SemChat is integrated with NEPOMUK through its PIMO component where the location, person and organization concepts are used to store the extracted concepts from a chat conversation. A better integration of SemChat with NEPOMUK's other components will be investigated in the future.

We opted to go for a multi-protocol based chat client rather then a single protocol such as Skype or MSN because this includes the possibility to connect to multiple chat protocols from within the same client. Spark IM[4] was found to be an ideal candidate for SemChat because apart from being open source, it could be further extended through plug-in development.

The extraction mechanism we opted for is based on XtraK4me[5] key-phrase extractor and ANNIE[6] named entity recogniser (NER), which is a component within GATE[7], since we require the extraction of the most important key phrases from a chat conversation and the identification of their entities. The main reason behind utilizing XtraK4Me was based on the fact that it makes use of several

[4] http://www.igniterealtime.org/projects/spark/index.jsp
[5] http://smile.deri.ie/projects/keyphrase-extraction
[6] http://gate.ac.uk/ie/annie.html
[7] http://gate.ac.uk/

GATE components and can also extract key phrases from both text documents and string representations unlike other key phrase extractors which were considered. On the otherhand, ANNIE NER is able to identify multiple entities and can also be extended to recognize user defined entities through JAPE[8], unlike other NERs considered.

Though various chat clients have a search facility, such as the case of Skype, this is limited in its capabilities. We intend to extend Spark's search facility to go over the extracted content and to allow for interesting searches such as searching by date or concept name to find any semantically related concepts.

2.1 SemChat's Concept Extraction Mechanism

When SemChat is enabled by the user, it monitors chat sessions and upon detecting the closure of a chat session or a chat room within Spark, SemChat starts its main processing. The reason behind the use of the end of chat session as a trigger for SemChat to provide useful information to the user, was mainly motivated by the requirement to implement the system as a non-intrusive one. By adopting this approach, SemChat in fact, strives to limit the cost of interruptions, which as described by [7] varies on average between 10 to 15 minutes before the users returned their focus to the disrupted task, which in this case would be the current chat activity.

SemChat starts by first retrieving the chat conversation between both users and passes this to the XtraK4Me key phrase extractor, which in turn identifies the main key words within a chat instance and finds the ones which are not already stored within the user's PIMO in NEPOMUK, by the use of NEPOMUK's search feature. All unique key phrases are then passed through ANNIE, so that their entities can be identified. ANNIE is able to recognize typical entities such as locations, persons, organizations and dates.

Once this process is complete the user is presented with a notification linked to a list of extracted concepts which is displayed in a separate tab. The intention behind this feature is to make the whole process less disruptive and distracting, as explained earlier. Context menus, as can be seen in Figure 2, are used to allow the user to choose to save a concept within the user's PIMO within NEPOMUK, thus confirming the importance and relevance of this concept, to delete a concept, indicating to SemChat that the concept is not relevant or to retrieve more information about the concept. In case the user chooses the first option, she can then also check that this concept was successfully stored within her PIMO and under the correct category. In case she wants more information about a particular concept, we have used Wikipedia[9] as our information repository, with snippets of information retrieved being displayed appropriately in a separate pop-up window.

The process of extracting possible events from a chat conversation is slightly different from that described above. In this case the whole chat conversation is

[8] http://gate.ac.uk/sale/tao/splitch8.html#x12-2080008
[9] http://en.wikipedia.org/

Fig. 2. A context menu showing the three options presented for each concept

passed directly through ANNIE to extract any existing events. Since by default ANNIE does not handle such entities it had to be extended. This was done by implementing a number of JAPE rules that specify how to recognize possible events within a chat conversation using regular expressions in annotations, as can be typically seen in Figure 3.

```
Phase: EventAnnotations
Input: Lookup DateClass
Rule: EventRule
(
  { Lookup.majorType==event_trigger }
):eventTrigger
-->
  {
   AnnotationSet matchedAnns= (AnnotationSet)bindings.get("eventTrigger");
   FeatureMap newFeatures= Factory.newFeatureMap();
   newFeatures.put("rule","EventRule");
   outputAS.add(matchedAnns.firstNode(),matchedAnns.lastNode(),
   "EventTrigger",newFeatures);
  }
```

Fig. 3. JAPE rule for annotating a sequence of text referring to a meeting

The implemented JAPE rules look up for different kinds of text sequences, such as phrases that may indicate a possible meeting, and different types of dates and time. Figure 3 shows the JAPE rule that was implemented to look up phrases which might indicate a possible event within a conversation. The *EventRule* rule will match any text that is an annotation of the *event_trigger* grammar. An *event_trigger* grammar consists of several phrases such as *"Meeting at"* and *"meeting with"* amongst others, which can all indicate a possible meeting. Once this rule matches a sequence of text, the whole sequence is allocated a label by

the rule, in our case this is *eventTrigger*. When this process is complete, any extracted events are also presented to the user in a separate tab.

For each extracted event, the user has the possibility to edit both the title of the event and also the prospective date details as are required. Any annotated event will automatically also be saved within Spark's Task List event scheduler as depicted in Figure 4. The user will be reminded of any forthcoming events by means of a notification on the event's due day.

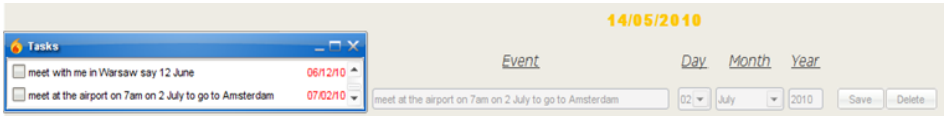

Fig. 4. The saved Event in the Spark's Task list event scheduler

Whenever a user switches off or logs out of Spark, any extracted concepts that were not annotated or deleted, within the current session, will be cached in a RDF storage. A list of such pending concepts is displayed to the user the next time that she enables the SemChat plug-in. We have implemented this feature in this manner so that the user would have *"another chance"* to annotate such concepts if deemed relevant. The concepts that are saved by the user are also cached in a separate RDF storage since they are used by the semantic search feature which will be discussed in the following section 2.2.

It is important to note that any deleted concepts are not cached, and they will be presented again to the user if they are extracted during another chat conversation. The reason behind this implementation is that some concepts, which are not deemed important during a particular chat, could still be seen as important during some future chat which has a different context. For example *during a particular chat session the name of David Guetta is mentioned. However at the time the user did not deem this to be important and deleted the extracted information. Nevertheless, during another chat conversation which was about the Isle of MTV show and which listed the said DJ as one of the participants, the user decided to annotate the person concept and find more information about it.*

2.2 SemChat Search

The semantic search feature helps the user to retrieve any of the annotated concepts. The user can filter-out a search by a number of defined criteria for example by date, whereby she will be returned with any semantically related concepts that satisfy these search criteria. This feature was implemented so that if a user needs to find some previous concepts, such as for example a previously annotated event, she can do so with ease, without the need to go through the whole chat transcripts. Each concept retrieved is presented to the user with its

full details and a typical example of a result obtained from the SemChat's search can be seen in Figure 5.

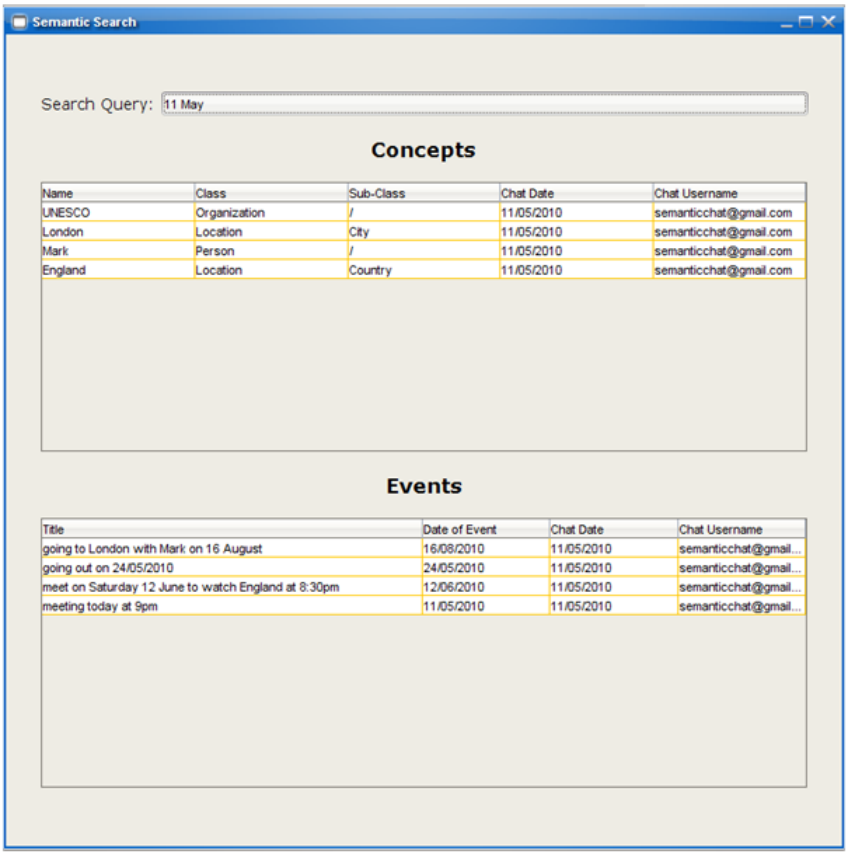

Fig. 5. Semantic search results

3 Evaluation

A usability session was organized as an initial effort to evaluate SemChat. In our setup we considered findings from previous research by [4] which outlined that 6-12 participants are enough to test the usability of a system and provide enough useful information such that initial but concrete conclusions can be made.

In line with this idea 8 participants, mostly students and colleagues, took part in this evaluation exercise. The evaluation session was split into three parts: the first part consisted in the exposure of SemChat's features through a walk-through

example; the second part involved each of the participants getting accustomed to SemChat by chatting with another participant for approximately 20 minutes; the third part consisted in each participant filling in a questionnaire which targeted several aspects of the system.

From the evaluation process, we were able to identify both the limitations as well as possible improvements that we could, in future, affect to our system. Based on the initial results, we could positively conclude that SemChats' main features of extracting concepts (and events) from a chat conversation and that of providing further information through Wikipedia, proved to be popular and useful features amongst the participants. The same can be stated for the integration of SemChat with Spark's event scheduler. It is important to note that the time that the extraction process takes depends on the length of the chat conversation since the more text there is, the more time it takes to extract the whole text. From the evaluation conducted, it was found that it took between 3 to 5 seconds to extract a conversation of approximately 20 minutes.

The semantic search feature was deemed to be less important by 50% of the participants, primarily because they did not find the need to search for any past annotated concepts. This is understandable, since the chat session was rather short. Yet another reason behind this could be attributed to the fact that participants were not accustomed to searching within chat conversations, since the majority of well-known chat clients, provide only limited search facilities, and thus possibly participants were unaware of the potential behind a semantic search facility. On the other hand, there was a high level of satisfaction amongst the other 50% of the participants who used the semantic search facility.

In some cases, however, important concepts flagged within a conversation were not extracted. We attributed this to the fact that the XtraK4Me key phrase extractor selects the most important key phrases according to their occurrence rate. In the future, this problem will be addressed by tweaking XtraK4Me.

It was also noted that in some cases, the event concepts were not being extracted, as expected. This was due to the fact that the events did not conform to the structure that SemChat's events extraction mechanism was implemented to recognize. An example of such an event was *"will be going to Holland"*, since no date or person's name was included in the phrase indicating such an event.

A possible solution for this limitation is to further extend ANNIE to recognize other different types of events that could be present within a chat conversation, however this might still not solve the problem completely. In [1] the use of *pidgin languages* is suggested to limit the different ways in which people record information in a note-taking tool, however this could be complicated to learn and at the end could possibly also be counter-productive.

In [2], the main problematic issues related to extracting information from chat are thoroughly analysed. Due to the *"noisy"* nature of chat content, in particular the fact that it may contain misspellings, non-standard use of orthography, punctuation and grammar, presents difficulties for generic information extraction engines. Furthermore the possibility of having *"interleaving of multiple topics and the effects of a dynamic, interactive mode of discourse where semantic*

content changes as the discourse progresses", makes it even more troublesome. The suggested solution, by [2], is based on a chat-specific, information extraction engine [13], that is capable of performing robustly when faced with such *"surface noise"* by typically allowing for chat data that contains non-standard orthography, punctuation, spelling and grammar.

4 Related Work

In this section we discuss some research which inspired our work on SemChat. The considered research is focused on the extraction of semantic information from notes and chat conversations.

ConChat [9] is a context-aware chat program which improves electronic communication by presenting contextual information. It tries to solve semantic conflicts which occur in chat conversations through the tagging of potentially ambiguous chat messages. ConChat solves part of this problem and therefore is a step forward towards eliminating semantic conflicts which occur in chat sessions. SemChat was designed in a way that it caters for some of the semantic ambiguities related to time, currency, units of measurements and date formats in a similar way to that in ConChat. In the case of time and date formats, this problem is catered in a different manner from ConChat since several JAPE rules were implemented to recognize different types of date formats that can be used within a chat conversation.

GaChat [6] on the other hand uses morphological analysis to extract the proper nouns from the dialogue text. Online images and articles from Wikipedia which are related in a way to these extracted nouns are simultaneously displayed alongside the dialogue text. This additional data is automatically displayed on the chat windows of both user and sender of the message to help reduce the elements of ambiguity like searching and also the asking of some particular details of a particular phrase. In the case of SemChat, the user has the option to seek further information from Wikipedia about each extracted concept.

SAM [5] tries to identify a number of problems that IM systems encounter in order to try to improve the content management of IM systems, moving towards the Networked Semantic Desktop. SAM extends a chat client by semantic annotations, semantic search, semantic browsing and semantic meta-data communication. SAM's chat window offers a taxonomy panel where the annotation of messages is permitted whilst a user is chatting. SemChat is similar to SAM, however in our case we extend Spark, which is also an XMPP protocol client, with the semantic annotations of concepts extracted from a chat conversation and with a semantic search feature based on the concepts that are annotated by the user. Nevertheless, within SemChat we store extracted concepts within NEPOMUK's PIMO and events are linked to an event scheduler, making SemChat more versatile and in line with PIM tools.

Though not directly related to semantic chat as the research mentioned above, Semanta[11] which is a plug-in to two popular email clients has some similarlities to SemChat which are important to mention. Firstly this system

uses the existing email transport technology and fully integrates with NEPOMUK. This is similar to SemChat, in fact the architecture behind our semantic chat client was inspired by Semanta. Secondly Semanta handles and keeps track of action items within email messages and also extracts tasks and appointments from email messages which are then added to the email client's scheduler. In a similar fashion, through SemChat it is possible to extract events from chat conversations which are manually annotated by the user and which are stored within Spark's task list scheduler. In this respect SemChat tags along the approach adopted by Semanta and not mearly adds a semantic component over the traditional chat component, as was mainly done in the research mentioned above, but strives to become a PIM tool in all respect.

5 Future Work

With regards to future work we have a number of interesting ideas, including the integration of SemChat with popular applications such as a popular email client like Thunderbird[10]. Through this integration any extracted events could be logged automatically into the email client's event scheduler, rather then keeping this information only available to Spark's task list event scheduler.

As already mentioned in Section 3, it is envisaged that other types of entities could be extracted from chat conversations, apart from the ones already identified. Typical examples of such entities could be, emails, products, addresses and telephone numbers. In this case, ANNIE would need to be further extended through JAPE in a similar manner adopted for events. The solution based on dedicated JAPE rules might however not always turn up each and every existing entity within a chat, due to the fact that chat data is inherently noisy, as explained in [2]. We are nevertheless confident that this approach complimented by user feedback can still achieve a satisfactory level of precision in identifying those concepts which are relevant for the user's PIM.

The semantic search feature could also be improved in several aspects. One such aspect is to further optimize the searching process since it has to sift through many annotated concepts and it takes some time to find all the semantic relations between the concepts satisfying the search criteria. The inclusion of an autocompletion facility, would make it easier for the user to retrieve the semantically related concepts in a faster and more efficient way.

This search facility could also be further enhanced such that it would display the part of the chat transcript from where each concept satisfying the search criteria was retrieved. Through this enhancement the user would be able to better recall the context within which a particular concept was mentioned during a chat conversation.

The semantic annotations generated by SemChat could also be quantitatively evaluated in the future. In this case the users could be assigned a set of tasks that will be conducted initially on a normal chat client and then performed also

[10] http://www.mozillamessaging.com/en-US/thunderbird/

on SemChat. This form of analysis might provide us with further insights into the costs and benefits of using such a semantic chat client for predefined tasks.

6 Conclusion

In this paper we presented SemChat, which is our initial effort at integrating a semantic chat component with a social semantic desktop, NEPOMUK. With the area of PIM increasingly becoming important, SemChat contributes further to this area through the integration of concepts in the user's PIMO as well as the integration of events with an events scheduler. Although the initial evaluation of the developed prototype is very encouraging, further work is required so that SemChat evolves into a fully realised PIM tool.

References

1. Michael Bernstein, Max Van Kleek, Mc Schraefel, and David R. Karger. : Evolution and Evaluation of an Information Scrap Manager. In CHI 2008 Workshop on Personal Information Management, Florence, Italy (2008)
2. Cassandre Creswell, Nicholas Schwartzmyer, Rohini Srihari: Information extraction for multi-participant, task-oriented, synchronous, computer-mediated communication: a corpus study of chat data. In Proc. IJCAI-2007 Workshop on Analytics for Noisy and Unstructured Text Data, Hyderabad, India, pp. 131-138 (2007)
3. Laura Dragan, Siegfried Handschuh. : SemNotes- Note-taking on the Semantic Desktop. In poster session of the 6th European Semantic Web Conference, ESWC'09, Heraklion, Crete, Greece (2009)
4. Joseph S. Dumas, Janice C. Redish. : A Practival Guide to Usability Testing (Revised Edition). Intellect Books, Exeter, UK (1999)
5. Thomas Franz, Steffen Staab. : SAM: Semantics Aware Messenger for the Networked Semantic Desktop. Koblenz-Landau, Germany (2008)
6. Satoshi Horiguchi, Akifumi Inoue, Tohru Hoshi, Kenichi Okada. : GaChat:A chat system that displays online retrieval information in dialogue text. In Workshop on Visual Interfaces to the Social and the Semantic Web(VISSW2009), Sanibel Island, Florida (2009)
7. Shamsi T. Iqbal, Eric Horvitz. : Disruption and Recovery of Computing Tasks: Field Study, Analysis, and Directions. In CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, San Jose, California, USA, pp. 677-686 (2007)
8. NEPOMUK How To. NEPOMUK Social Semantic Desktop. http://dev.nepomuk.semanticdesktop.org/wiki/UsingNepomuk (2008)
9. Anand Ranganathan, Roy H. Campbell, Arath Ravi, Anupama Mahajan. : ConChat: A Context-Aware Chat Program. IEEE Persuasive Computing, Vol. 1, Issue 3 (2002)
10. Simon Scerri, Brian Davis, Siegfried Handschuh, Manfred Hauswirth. : Semanta - Semantic Email made easy. In Proceedings of the 6th European Semantic Web Conference, ESWC'09, Heraklion, Crete, Greece, pp 36-50 (2009)
11. Simon Scerri, Ioana Giurgiu, Brian Davis, Siegfried Handschuh. : Semanta - Semantic Email in Action. In Proceedings of the 6th European Semantic Web Conference, ESWC'09, Heraklion, Crete, Greece, pp 883-887 (2009)

12. Michael Sintek, Siegfried Handschuh, Simon Scerri, Ludger van Elst. : Technologies for the Social Semantic Desktop. In Reasoning Web. Semantic Technologies for Information Systems: 5th International Summer School 2009, Brixen-Bressanone, Italy (2009)
13. Rohini K. Srihari, Wei Li, Cheng Nium, Thomas Cornell. : InfoXtract: A Customizable Intermediate Level Information Extraction Engine. In Journal of Natural Language Engineering, Cambridge U. Press, 14(1), pp.33-69 (2008)