A FRAMEWORK FOR BUILDING
ADAPTIVE INTELLIGENT VIRTUAL ASSISTANTS

Luc Lamontagne¹, François Laviolette¹, Richard Khoury², Alexandre Bergeron-Guyard³
¹ Department of Computer Science and Software Engineering, Université Laval, Québec, Canada
{Luc.Lamontagne, Francois.Laviolette}@ift.ulaval.ca
² Department of Software Engineering, Lakehead University, Thunder Bay, Canada
Richard.Khoury@lakeheadu.ca
³ Defence Research & Development Canada, Valcartier, Canada
Alexandre.BergeronGuyard@drdc-rddc.gc.ca

ABSTRACT
This paper describes a framework to support the construction of intelligent virtual assistants (IVAs). An IVA agent is a software assistant capable of interacting with a user to support sense-making tasks, to determine information needs, to provide relevant information and to improve its performance based on user feedbacks. Currently, there is no integrated software environment available to develop such agents. We are exploring how we can integrate machine learning and natural language processing technologies, available as open source software, to support the construction of intelligent virtual assistants. The framework relies on a combination of question answering (Q/A), information extraction (IE) and user modeling components. In this paper, we present an overview of the work that is being conducted to build a prototype of the framework.

KEY WORDS
Intelligent virtual assistant, question answering systems, information extraction, topic modeling, machine learning, natural language processing.

1. Introduction

The motivation for this research effort is the development of an Intelligent Virtual Assistant (IVA) capability for supporting the intelligence community in their sensemaking tasks. The design of such systems relies on intelligent personal assistant technology [1], i.e. software agents capable of performing tasks with minimum guidance from its users. Intelligent virtual assistants combine artificial intelligence technologies to support users, to organize information and to adapt to changing situations.

The field of intelligent assistants is currently an active area of research and development with some products being much publicized:
- CALO and PAL were projects lead by the Stanford Research Institute (SRI) to investigate how machine learning can help to design software agents performing in an office automation environment. Various modules were developed during this project for categorization and filtering of RSS feeds (C2RSS), recommendations of documents (PrePak), auto-completion of forms (FOAM) and activity recognition (Warp).
- Siri, a spin-off of the CALO project, is a voice-activated personal assistant commercialized by Apple to provide factual information about various information products such as movies, restaurants, weather reports and concerts.
- Google Now is a recent effort by Google to provide services on their Android operating system to deliver factual information on a limited number of topics such as weather reports, flight information, restaurant reservations and movies.
- Mitini and S-Voice are Siri-like systems developed by Microsoft and Samsung.

All these systems are proprietary and limited description of the underlying technology can be found in the literature. They are targeted for the general public, hence aiming to emphasize ease of use. Their capabilities are designed to support tasks requiring minimal domain training such as personal data management and product recommendation. Interactions with these personal assistants rely on voice-user interfaces mostly leaving the initiative to the users.

As part of our research project, we investigate technologies to design intelligent virtual assistants for specialized users, i.e. users performing organized tasks on a recurrent basis. Our main application domain is to assist sense-making tasks such as those conducted by military intelligence analysts. Taking information provided by incoming news reports and military messages, an intelligence analyst must produce various intelligence reports. The category of IVA systems targeted in this project is an interactive system helping an analyst to retrieve, extract and reuse relevant information from various textual and semi-structured sources.

Such IVA systems have to cope with the following particularities:
- **Task-oriented information search**: Tasks such as the production of intelligence reports follow some specific procedures and domain models that can be exploited to determine the type and nature of information required at each step.
• **Restricted domain:** situations analyzed by intelligence analysts take place in limited geographical environments, involving targeted objects/units and relating to specific topics of interest. Information about the operational context can also be exploited to improve the relevance of system assistance.

• **Specialized users:** analysts have a high level of training and deep knowledge of their application domain. Hence they have specific information needs in specific situations. And they can provide useful feedbacks on the relevance of system recommendations and initiatives. More importantly, they can demonstrate to the system how to cope with complex situations.

Our approach is to exploit open source technologies to develop an intelligent software assistant framework based on natural language processing (NLP) and machine learning (ML) techniques. And we are validating this technology on the description of military situations, an application domain requiring richer background knowledge than product recommendations or personal data management. We give an overview of these research efforts in the next sections.

2. **Context for IVA**

This work is conducted in the context of a Defence Research and Development Canada (DRDC) project called Intelligence Virtual Analyst Capability (iVAC) [2] which is an intricate part of the Future Intelligence Analysis Capability (FIAC) [3]. The iVAC project investigates and develops ISA technologies towards the development of a computerized software assistant supporting the intelligence analysts in sensemaking tasks, while being ultimately capable of taking on autonomous analytical tasks in concert with other analysts (virtual or human).

As part of this research, an identification of iVAC sub-capabilities was performed, based on literature reviews and workshops held with experts from the military, the industry, and academia [4]. This work allowed identifying question answering and dynamic user modeling as component of high value for the development of an iVAC. The IVA project described in this paper is directly aligned with iVAC priorities and contributes directly to the development of a system that adapts to the user's context, and answers questions based on context and available information.

3. **Components of the IVA framework**

The choice of components to include in the IVA framework was guided by different motivations. We present in the next paragraphs, a detailed description of the components we selected to be part of the framework.

3.1 **Integrated framework for the design of IVAs**

As illustrated in Figure 1, an IVA aims to support a user in its domain specific tasks. For our application domain, an IVA helps an intelligence specialist to analyze a situation involving different aspects including geopolitical, socioeconomic and demographic factors.

As most of the information sources available in our application domain are textual in nature, the IVA framework includes a question-answering component that can be queried by the user to gain information about some of these factors. Typical queries would pertain to:

- Characteristics of domain entities such a military units, ethnic groups, political actors, infrastructures and important dates.
- Relations such as roles of leaders, population estimates of some regions, nature of conflicts between factions, electricity or natural resources providers, etc.
- Events depicting actions taken by various actors present in the region of interest.

To facilitate the search for relevant information passages, an information extraction component processes domain-related documents (ex. new reports and military messages) and annotates them with domain specific tags. We expect that the insertion of annotation scheme will improve the relevance of system responses. Categories of annotations targeted in the project are named entities and semantic relations. Some annotations are generic (ex. named entities) while some specific annotation sets could result from information needs of the user.

As the technology is targeted for specialized users, we want to personalize the recommendations of the IVA system by constructing user models from unsupervised machine learning techniques. Construction of the user models rely on the analysis of traces depicting the information and documents manipulated by the user. Activity traces contain a combination of the following information:

- Documents received to the user from external sources. In our current experimentation, most of the external sources are RSS feeds that are managed using RSSOwl as a news feed reader;
- Documents consulted by the user from local collections or external sources;
- Queries submitted by the user to the IVA system;
- Inputs provided by the user in the textual fields of intelligence templates (i.e. some reports produced by the analyst).

User models are the pivot elements of the IVA framework as they capture the potential information needs by the user, support the analysis and the disambiguation of the questions submitted to the Q/A component, and provide guidance for the annotation of documents in the local collections. User models can also be augmented through explicit interactions with users.
The three components of the framework are further described in the next sections. Moreover, we then explain some of the interactions taking place among these components.

3.2 Automatic annotation of textual documents

Searching for answers in textual documents can be facilitated if the type of information being sought is clearly highlighted. For example, one might be searching for some specific entities (ex. individuals, organizations, locations) or some relationships between individuals and events. By performing *a priori* annotation of documents as a batch process, one would expect to increase the relevance of the passages returned by the Q/A system without impacting on response time.

The text annotation component of the IVA framework includes two types of information extraction systems. First a named entity recognition (NER) system [5] is trained to detect the domain entities of interest to the user. Our implementation relies on the Stanford NER libraries [6] that can be trained with tagged data sets. NER technology is mature [7] and provides satisfying results if combined with domain-specific resources (ex. gazetteers and name lists) and representative training data.

Second, we extract semantic relations from texts. It consists of finding triplets of the form “*x relation y*” that are recurrent structures in textual documents. For instance, the sentence “*Barrack Obama is the president of the United States*” would be decomposed as a triplet containing the two arguments “*Barrack Obama*” and “*United States*” and the relation “be the president of”. Significant progress has been made recently by open information extraction systems on the extraction of meaningful relations out of documents [8, 9, 10, 11]. Some algorithms are showing good results for identifying relation phrases on a large-scale corpus without any specific domain training other than using simple linguistic patterns. Recent algorithms extend relations extracted by these systems to phrases mediated by verb and noun patterns. Moreover, these systems offer the advantage of not requiring any pre-specified vocabulary or set of relations. Hence they can be used for the discovery of important relations to be subsequently validated by domain experts.

If more accurate extraction capabilities are required, the extraction process could be augmented by semantic role labelling algorithms [12] that associate verbs to a semantic structure and map the other phrases of the sentence to roles in this structure. Supervised machine learning techniques (such as SVM or MEMM classifiers) can be used to tag text documents with semantic classes and roles relevant to the application [13, 14]. However we do not pursue this approach at the moment due to the good results obtained with the information extraction components (see Section 4 of this paper).

3.3 Learning of User Models

Learning a user model amounts to capture the potential information needs of the users. For instance, an IVA for intelligence operations should have some representation of the military, geopolitical and economical factors that might be required at various stages of operations. An IVA system must be able to learn these models from user’s feedbacks, together with past questions and prior manipulations of documents. Such models can be exploited to improve system’s performance.
In our current framework implementation, user models are constructed using topic-modeling algorithms [15]. Topic modeling tools are unsupervised learning algorithms that represent topics as probability distributions over various clusters of words (the topics). We make use of the MALLET library that includes algorithms such as Latent Dirichlet Allocation (LDA), hierarchical LDA and Pachinko allocation [16, 17]. Topic models are constructed from the activity traces provided to the framework by the RSS news feed reader.

Taking advantage of the information extractors of the framework, we include in the topic models extracted roles such as named entity classes and relation phrases. These augmented topics are useful as they can capture high-level interests of the users. For instance, a model could contain individual terms such as “negotiation” and “talk”, roles such as “ORGANISATION” and relations such as “be scheduled”.

While this has not been undertaken yet, we would like to model the context in which the user needs specific information. To acquire these context descriptions, monitoring procedures have to be developed for acquiring demonstration traces from users. In learning by demonstration paradigm [18], a system learns how to behave from examples provided by human teachers. The demonstration consists of examples provided as sequences of state-action pairs, indicating to the system how a task should be accomplished [18,19,20,21]. For instance, in our application, demonstrations depict the information targeted by intelligence analysts at the various stages of their analysis and the textual passages inserted in the intelligence reports. From the demonstration traces, algorithms such as those used for on-line topic detection and tracking [22] could be used to capture context episodes and to reuse them as guidelines when facing new situations.

3.4 Adaptive Question Answering

As a virtual assistant is mainly an information service, the main functionalities of the IVA framework are provided by a question answering (Q/A) system. The modules encountered in the pipeline of a Q/A system [23] are implementing three types of functions: question analysis, passage retrieval and answer selection.

Question analysis consists of understanding what is being asked by the user. At a minimum, the system needs to classify the topic of the question and to extract important keywords featured in the question. Additional steps are also added to determine the expected answer type such as factoid information, opinions or short definitions. It also provides information on the kind of information requested (ex. a person, a location, a yes/no answer). Accurately pinpointing the characteristics of answers can greatly improve the accuracy of a Q/A system, notably by making possible to use type-specific or kind-specific information retrieval algorithms and by filtering out answers of the wrong type [24].

Passage retrieval retrieves and filters out text fragments that might contain the answer sought by the user. A typical Q/A system must include an information retrieval (IR) subsystem to search for documents relevant to the question asked, as well as an information extraction (IE) subsystem to pinpoint relevant passages within longer texts. In some Q/A systems, these two steps can be combined. For example, we can break up a long document into parts (such as individual paragraphs) and allow the IR subsystem to retrieve these individual parts. While creating a simpler architecture, this allows the system to easily account for long documents discussing multiple different topics in turn. However, we kept these as two different steps in our Q/A system. Many different IR algorithms have been proposed in the literature to handle the different information sources. They can range from simple keyword searches of plain text documents to collaborative filtering of online resources. Likewise, the IE step will vary depending on the knowledge base being used. Simpler systems can operate at the word level, for example by returning fields with the correct keyword label in semi-structured documents or sentences featuring named entities found in the question. More sophisticated systems often operate at the sentence-syntactic level, aiming to discover text fragments that do not simply mention the correct keywords of the question but mention them with the sentence structure of the answer that is expected given the structure of the corresponding question [25].

Answer generation presents the answers in a format acceptable to the user. The passage retrieval phase returns several relevant text fragments. In its simplest form, a Q/A system ranks these fragments and returns the ranked list to the user. Several improvements are applied on this basic strategy, such as grouping together duplicate answers from different text fragments, or presenting text fragments with their surrounding sentences to capture the context. More advanced systems will actually rephrase the information discovered into an original answer statement. Some systems use semantic templates that specify which answer component to use in a given field [26], while others prefer syntactic templates that specify word order in the answer [27]. In addition, some Q/A systems implement different templates for the different devices. For example, a verbose template style would be used when displaying results on a desktop computer, while a concise template that omits unnecessary contextual information would be used when displaying the result on the smaller screen of a mobile device [26].

Building on this three-step pipeline architecture, an adaptive question answering system [28] is a Q/A system that can improve its performance based on the operational context. Q/A systems can be made adaptive by injecting models describing the goals and preference of the questioner and the characteristics of the task being performed. Adaptive Q/A has not been the subject of extensive research, as the recent trend has been to have stateless systems where users ask only one question per session [28]. This is a trend justified by studies showing that this is the way a majority of users behave, especially
when it comes to using online query systems [29]. However, this differs in the application domain targeted by our project. The same specialized users query the Q/A component of the IVA framework repeatedly for consistent purposes. Such system can clearly benefit from adapting its behaviour to its users and purposes.

The adaptive Q/A component of the IVA framework is built on Open Ephyra, an open source Q/A infrastructure. Currently, information provided to the system is taken from Wikipedia, the CIA World Factbook and also includes all the documents received from subscribed RSS news feed. Lucene and Indri search systems are responsible to index the RSS news reports in the local collections and to support the passage retrieval phase of the Q/A system. More specifically, full documents are first searched using Lucene. Passages are then extracted using both Lucene and Indri snippet extraction capabilities. This choice is mainly due to the complementary nature of the passages extracted by both systems and to the greater stability of Lucene for indexing large collections of documents.

The IVA framework extends the Q/A infrastructure with techniques exploiting the user models acquired from user’s activity traces (these techniques are described in Section 5). Our aim is to go beyond answering explicit factoid questions, and to return information that is relevant to the user asking the question within the context of the user model. Techniques combining information from the user model, from the system’s internal annotated knowledge base and from external knowledge bases (such as Wikipedia) can be used to improve the performance of the three steps of the Q/A process.

4. Preliminary experiments

We conducted some preliminary experiments early in the project to determine the baseline performance of the IVA framework for our military intelligence application domain.

The annotation component relies on two extraction technologies. We first evaluated the NER extraction component that we have selected for the project (Stanford NER). We obtained accuracy greater than 80% on a corpus of geopolitical news reports representative of our military application domain. We estimate that these results fully meet our expectations. However additional training could be performed on our domain corpus if higher levels of accuracy are required for some other application domains.

We also evaluated the performance of REVERB [6] as relation extractor for our framework. Experimentations conducted on our corpus of geopolitical news reports resulted in an accuracy of approximately 60% of the relations present in these documents. Moreover, most of the relations extracted by the system were valid and could be made to contribution in the framework. This result suggests that most of the important relations of an application domain can be identified and exploited as part of the Q/A process.

The performance of the Q/A component was evaluated on different sources of information. To do so, we built a list of 200 questions that were submitted to the system. Most of the questions were related to various geopolitical conflicts occurring during the months our experiments were performed (ex. Syria and Mali). We made sure that all the answers were present, in some forms, in the corpus or resources exploited by the information sources. Hence our goal was to assess the capability of the Q/A to locate, circumscribe and rank the relevant text passages knowing that these could be provided by the information sources.

We estimated from our experimentations that, overall, less than 30% of the questions were correctly answered by the most highly ranked passage of the Q/A system. The fragmented results for each information source are presented in Table 1.

Questions that were considered to present either low or average level of difficulty (ex. simple factual information and definitions) were found in a reasonable proportion by the system (between 30-40%). However, none of the challenging questions considered as difficult or very difficult could be answered by the system.

Table 1

<table>
<thead>
<tr>
<th>Preliminary Q/A results for different sources of information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Wikipedia &amp; CAI World Factbook</td>
<td>35%</td>
</tr>
<tr>
<td>Domain corpus with Indri search engine</td>
<td>25%</td>
</tr>
<tr>
<td>Domain corpus with Lucene search engine</td>
<td>22%</td>
</tr>
<tr>
<td>Domain corpus with Indri and Lucene search engine</td>
<td>24%</td>
</tr>
</tbody>
</table>

However, following a detailed analysis of the Q/A results, it appears that these baseline results can greatly be improved since more that 50% of the answers were contained in the first 5 passages returned by the system. And over 90% of the answers were present in the tens of passages extracted by the system. Hence providing better answer selection schemes could significant improve the accuracy of the system. We explore in the next section some research avenues we are currently pursuing to improve these results.

5. Adaptation of IVA behaviour

Following the preliminary experiments that we conducted on our military intelligence application domain, we identified candidate techniques in order to be added to help adapting the IVA framework to new users and new application domains. Three of them, taking advantage of synergy between the components in the framework, are described in the next paragraphs.

5.1 Questions expansion and paraphrasing

A common technique to improve the performance of Q/A and IR systems is query expansion, which consists of supplying common domain-specific keywords missing in the question. The need for this step stems from the fact that most questions are very short, between one and four
keywords long on average [30] (after removal of the words bearing no specific meanings). Moreover, these keywords can have ambiguous meanings [31]. These two factors limit the accuracy of any information retrieval operation based on short question. Query expansion can benefit to the system by allowing it to retrieve documents and answers that would otherwise be missed because they do not have any keywords in common with the original question [32]. More specifically, we design the question analysis phase of the Q/A system in such a way that it includes a step where the question is rewritten into multiple paraphrased forms. This step allows the system to recognize different forms of the same question and then to expand them all so to improve the recall of text passages by the information retrieval system.

In our current work, we are exploring how questions can be expanded based on the paraphrasing of their main relation. For instance, we would like to expand the question “When was Louisiana acquired by the USA” into multiple paraphrased versions using relations such as “bought by”, “purchased by” and “became part of”. To do so, we need to create a linguistic resource that provides clusters of relations based on synonyms, called synsets. As such a resource is not currently available, we have built a preliminary version using relations extracted from Wikipedia with the text annotation component of the framework. Given a list of relations and the context in which they appear, we can statistically evaluate the semantic relatedness of the relations. The context of a relation is built from the list of words surrounding its occurrences in sentences. And the semantic similarity of two context vectors can be estimated using similarity measures (we chose a normalized cosine function). For instance, the synsets obtained with this scheme for the relation “announce ceasefire” is:

“[“announce victory” (0,316) “declare neutrality” (0,258) “announce settlement” (0,223) “declare ceasefire (0,223), ...]”

As tens of thousands relations can be mutually compared for a single application domain, this process can lead to a computational explosion. We reduce the complexity of the learning scheme by using Wordnet to limit the comparisons to relations containing related words. A simplified version of the pseudo-code for the overall approach is provided in Figure 2.

```plaintext
function SELECT-FILTERS(filters, questions, answers)
1. Generate an initial array of populations containing different combinations of filters;
2. Evaluate the fitness of each combination using the training pairs of questions and answers;
3. Select at random (using a Russian roulette function) two combinations for each population;
4. Swap parts of the selected combinations using a one-point crossover function and add them to the population;
5. Mutate some filters in the newly generated combinations;
6. Migrate some combinations from one population to the other
7. Select a number of combinations to remove (equal to the number of new combinations);
8. Repeat from step 2 until some stopping criterion is satisfied;
9. Return the combination of filters having the best fitness value in the final populations

Figure 3. Pseudo-code for the selection of filters
```

### 5.2 Adaptive filtering and scoring of the answers

As mentioned previously, we estimated from our experimentations that less than 30% of the questions were correctly answered by the most highly ranked passage of the Q/A system. However, this could greatly be improved since more than 50% of the answers were contained in the first 5 passages returned by the system. Hence, providing better answer selection schemes could significantly improve the accuracy of the system.

Answer selection consists of applying a series of filters to rate passages based on their textual content and to remove those that do not match the characteristics of the questions. Hence, the quality of the results highly depends on the choice of filters included in the pipeline and on the order in which they are placed.

To address this problematic, we have implemented a genetic optimization algorithm that seeks for the optimal configuration of filters to be included in the system. A genetic algorithm [33] is a search technique that uses specific heuristics, inspired from the process of natural selection, to determine how to conduct its optimisation. The optimization routine consists in generating an initial population of individuals (representing potential solutions to our problem) and making this population evolve over time by modifying the characteristics of the individuals.

```plaintext
function BUILD-RELATION-SYNSETS(relations, triplets)
for every relation r1 in relations
    merge WordNet synsets for all the words in r1 as syn1.
for every other relation r2 in relations
    retain as candidates1 the relations r2 containing words from syn1.
for every relation r1 in relations
    for every other relation r2 in candidates1
        build context vectors v1 and v2 from triplets of r1 and r2.
        compute similarity s between v1 and v2.
        retain the top relations relation_synset1 ranked by decreasing similarity s.

Figure 2. Pseudo-code for building relation synsets
```

In our application, each individual corresponds to a sequence of a subset of the filters. The characteristics of each individual indicate the relative importance of a specific filter. To conduct the optimization, a routine has also been implemented to evaluate the fitness of a combination of filters on a set of training questions and answers. For Q/A, the fitness of combination of filters corresponds to the accuracy of the answers selected by the filter combination. Hence given a list of question-answer pairs provided by the user, it is possible to customize the answer selection phase of the Q/A system to the specific usage of the system by some users. The pseudo-code of the optimization routine is described in Figure 3.
5.3 User-specific ranking of the answers

To exploit user models as part of the question answering process, we also included a probabilistic scoring approach based on the topics for the questions and the candidate answers. First, the set of relevant topics to the user is learned based on a topic modeling approach such as Latent Dirichlet Allocation (LDA). The models are learned from the news reports being read as well as from the questions being asked by the user. Using these models, it is possible to estimate the similarity of a new question asked by the user to the candidate answers returned by the system.

The score of a candidate answer, i.e., a text fragment returned by the passage retrieval phase of the Q/A system, can be estimated as the highest probable topic given some answer and some question. More formally, the topic-model score of a candidate answer can be evaluated as:

\[
\max_{t_a} p(t_a|q, a)
\]

where \(t_a\) is the answer topic, \(q\) is the question formulated by the user and \(a\) is a candidate answer returned by the search engine.

Equation (1) can be further decomposed, using Bayes and chained rules, as:

\[
\max_{t_a} p(t_a|q, a) = \max_{t_a} p(q|t_a)p(a|t_a)p(t_a)
\]

which corresponds to the multiplication of three probability values designating:

a) The relatedness of the query to the topic,

b) The relatedness of the answer to the topic, and

c) The likelihood of occurrence of the topic in the Q/A process.

All three probability distributions can easily be obtained from a topic-modeling algorithm applied to the information exchanged between the IVA system and the user. We are currently performing an evaluation of this answer-filtering scheme.

6. Conclusion

This paper gave an overview of the efforts we are devoting to the development of a framework for building intelligent virtual assistants. We described the three main components of it and explained how they are in the framework. Moreover, we discussed some of the practical issues we are facing. We have discussed three adaptation techniques that are being added to the framework in order to improve its performance. Details were provided on technical decision made during the implementation. Future work on this project will include a global evaluation of the IVA system including the adaptation scheme presented in Section 5 of the paper. And further experimentations will also be required to explore how to build user models that are not strictly constructed from topic modeling algorithms.

Acknowledgement(s)

We would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canadian Department of National Defence for the financial support of the work through their joint DND-NSERC research program.

References


