Shoo the Spectre of Ignorance with QA^{2}SPR

An Open Domain Question Answering Architecture with Semantic Prioritisation of Roles

Simone Scannapieco[1] and Claudio Tomazzoli[2]

[1] Real T R&TD Department, Verona, Italy
simone.scannapieco@realt.it
[2] Parco Scientifico e Tecnologico, Verona, Italy
claudio.tomazzoli@parcoscientificoverona.it

Abstract. Open Domain Question Answering (ODQA) aims at automatically understanding and giving responses to general questions posed in natural language. Nowadays, the ability of a ODQA system is strictly dependent on how valuable information is effectively discovered and extracted from the huge amount of documents on the net – may it be structured (e.g., online datasets), or unstructured (e.g., free text of generic web pages). This, in turn, relies on a proper (i) identification of question keywords to isolate candidate answer passages from documents, and (ii) ranking of the candidate answers to decide which passage contains the correct answer. In this paper we introduce a Question Answering Architecture with Semantic Prioritisation of Roles (QA^{2}SPR) where a novel technique of *prioritised semantic role labelling* (PSRL) is used to optimise such phases. We also share the experimental results collected from a working prototype of QA^{2}SPR for the Italian language.

1 Introduction and Motivations

Question Answering Systems (QAS) are particular types of Information Retrieval (IR) systems that process user queries (*questions*) posed in natural language and retrieve the closest or correct amount of information required by the query (*answer*). Since the development of first restricted domain QASs, the scientific community has witnessed a widespread interest about QA-related topics. Only in the last two decades – in particular the period 1999-2007 – the body of literature in the field has grown so large and diverse that it is extremely difficult to survey all research areas stemmed from this discipline (IR, Information Extraction, Natural Language Processing, and many others). An exploratory analysis has shown that the number of surveys, reviews, and conference papers on the subject has increased by a factor of fifteen from the period 1960–1999 to 2000–2017 [1,12]. The motivation behind such phenomenon is twofold. On one hand, QA tracks of annual conferences like TREC, CLEF and NTCIR contribute to maintain a stimulating and challenging research environment over the years [1]. On the other hand, the exponential growth rate of digital data (such as the number of web pages on the Internet grown from 200 billion in 2006 to over 1 trillion in 2008 [14]) allowed to access a massive pool of information and model highly sophisticated QASs that answer more and more complex user questions (e.g., definitional questions, list questions, or why-type questions). This factor plays a key role in building deep interrelations between QA research and Knowledge Discovery (KD), whose major aspect is to extract valuable knowledge and information from web data.

Nevertheless, treating such amount of data also means to tackle several hidden pitfalls that may threaten QA sub-task performances. Specifically, allowing more complex user questions makes more difficult for the system to determine the expected answer type, i.e., to classify the answer based on the category of the subject required by the question (*question classification* phase). As an example, we expect that the answer to "Who invented the light bulb?" regards a *person*. Moreover, both *document processing* phase – i.e., the keyword-based retrieval of web documents as much pertinent as possible with the answer topic – as well as the *answer processing* phase – i.e., returning a ranked list of candidate answer passages extracted from such documents – are extremely prone to errors in scenarios characterised by high volumes of available information.

Question classification, document processing, and answer processing are clearly crucial to extract correct and precise answers. We cite, among others, the error analysis of a ODQAS by [10] which shows that more than 30% of wrong answers are due to incorrect question classification. However, while question classification research has already produced very satisfying, quite definitive results (e.g., classification accuracy of up to 90% [14], or even better [9]), the vast *plethora* of plausible heuristic metrics and algorithms that can be used to afford the other two phases make research still wide open to new proposal and improvements in this direction.

In the present paper we introduce QA^{2}SPR (Question Answering Architecture with Semantic Prioritisation of Roles), an ODQA system architecture that exploits a novel technique of question analysis in natural language – called *prioritised semantic role labelling* (PSRL) – aimed at optimising question keyword extraction, document processing, and answer processing phases. Moreover, we present an embodiment of QA^{2}SPR through a working prototype for the Italian language.

The paper is organised as follows. Section 2 briefly reviews the QAS topics that drove QA^{2}SPR design, and stresses out the novelty of our work w.r.t. existing semantic role labelling methodologies. In Section 3 we delineate the operating principles of PSRL by means of examples. Section 4 reports a brief description of QA^{2}SPR and the most important building blocks we adopted for its realisation. Section 5 reports the experimental results by a working prototype of QA^{2}SPR for the Italian language, while Section 6 draws some open issues left for future work.

2 Related Work

The theoretical work behind QA^{2}SPR architecture design and the prototype realisation are the result of a meticulous and critical study of answer extraction techniques in factoid question answering [19], semantic approaches for question classification [14], and ODQA based on syntactic and semantic question similarity [6], with special consideration of existing QA system implementations [13, 20].

Although QA^{2}SPR strictly follows the dictates of current state-of-the-art methodologies existing in the literature, the main novelty of this work rather relies on the strategy we devised for user questions analysis, that is, PSRL. Our technique takes mainly inspiration from *frame semantics theory* and *semantic frame* representations [7]. The semantic frame approach identifies the meaning of words through schematic representations of the situations that characterise human experience, each constituted by a group of participants in the situation, or *frame elements* (FEs), and describes the possible syntactic realisations of the FEs for every word. Usually, the information necessary for the individuation of semantic frames is gathered by annotating (*labelling*) *corpus* sentences in a specific language with FEs (*semantic roles*) and syntactic informations. Several (semi-)automatic techniques frame extraction from real world *corpora* (e.g., the British National Corpus) gave rise to popular online resources such as FrameNet, PropBank, and WordNet. It is nowadays widely acknowledged that linguistically annotated *corpora* have a crucial theoretical as well as applicative role in NLP, and QA is often cited as an obvious beneficiary of semantic role labelling. WordNet has been already profusely employed in QA-related tasks ranging from query expansion, to axiom-based reasoning, passage scoring, and answer filtering, while syntactic structure matching has been applied to candidate passage retrieval and answer extraction (see [17] for a complete list of references).

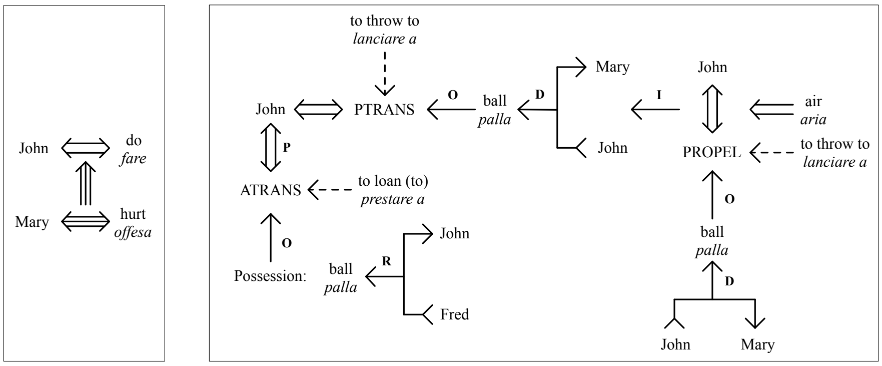
The ever growing popularity of semantic role labelling applied to question analysis persuaded us to choose a similar approach in our system as well. Nevertheless, although QA^2SPR has been designed as a modular architecture configurable with an arbitrary language, our first case study envisaged a QA tool for Italian,[3] which still lacks stable or well documented semantic annotated resources [8, 11, 18]. We then opted to model PSRL as a novel semantic frame-like approach for Italian *logical complement analysis* based on *Schank verb theory* [16]. In the same way frame semantics gives an heuristic model to isolate relevant frame elements based on *corpora* annotations and schematisations of real world situations, we argue that Schank verb analysis may give an heuristic model to isolate logical complements from questions based on semantic verb content. Elevating the importance of verbs to grasp the meaning of the whole sentence is in line with [8], where authors state that a more rigorous and clearly defined methodology for the study of verb semantic distribution is mandatory when coping with complex languages like Italian. Conversely, annotated resources such as FrameNet for English do not take into account the general distribution behaviour of a verb, nor it is even represented within the standard FrameNet format for Lexical Units (LU). In Section 3 we shall briefly delineate the theory underlying PSRL. For space reasons, what follows is the description of a simplified version of the real implemented procedure, but complete enough for the reader to capture all the basic working principles.

3 PSRL: Theory and Examples

Schank verb analysis [16] maps natural language utterances into conceptual structures that are unambiguous representations of their meaning, *independently from the language used*. A *conceptual dependency framework* (or *conceptualisation*)

[3] In the following, all references and examples in Italian language shall be reported in italics with the corresponding English translation in regular typeset.

is devised, which models two basic constructions: (i) actor-action-object – e.g., "John$_{\text{actor}}$ hurts$_{\text{action}}$ Mary$_{\text{object}}$" ("*John*$_{\text{actor}}$ *ha offeso*$_{\text{action}}$ *Mary*$_{\text{object}}$") – and (ii) object-state – e.g., "Mary$_{\text{object}}$ is hurt$_{\text{state}}$" ("*Mary*$_{object}$ *si é offesa*$_{\text{state}}$"). The combination of the two permits to paraphrase an arbitrary sentence so as to explicit the actual conceptual relationships. For instance, the construction "John$_{\text{actor}}$ hurts$_{\text{action}}$ Mary$_{\text{object}}$" violates the rule that conceptual actions must correspond to real world actions: the verb "hurt" does not refer to any action that actually occurred, but rather to the result of the action that actually occurred (which is unknown). Thus, the sentence should be rephrased as "John$_{\text{actor}}$ does$_{\text{action}}$ something that causes Mary$_{\text{object}}$ to be hurt$_{\text{state}}$". A graphical represen-



(a) (b)

Fig. 1: Schank graphical representation: (a) Simple scenario; (b) Complex scenario.

tation of the utterance is reported in Figure 1(a), where $\Longleftrightarrow$ denotes the mutual dependency between actor and action, $\Uparrow$ denotes the *causal* relationship (i.e., Mary was hurt *because* John did something to her), and $\Longleftrightarrow$ indicates the object-state complex. Schank theory postulates that only fourteen *action types* – falling into four distinct categories, namely Instrumental, Physical, Mental, and Global – suffices to conceptualise arbitrary complex statements in natural language. Consider the sentence "John threw the ball (that) Fred loaned him to Mary" ("*John lanció a Mary la palla che Fred gli aveva prestato*"), depicted in Figure 1(b). Schank analysis unravels the utterance as follows: the *actor* "John" performs an action that constitutes a *change in physical location* (Global action type PTRANS inferred by verb "to throw to", "*lanciare a*" – ⇠) of *object* "ball", "*palla*" ($\xleftarrow{o}$); the *direction of physical change* (**D**-labelled) is from the *donor* "John" to the *recipient* "Mary". The *instrument* used to cause PTRANS ($\xleftarrow{I}$) is *acting by propelling* (Physical action type PROPEL inferred again by verb "to throw to") the *object* "ball", usually made through the *medium* "air", "*aria*" ($\Longleftarrow$). Moreover, there has been a *change in the abstract relationship* (Global action type ATRANS inferred by verb "to loan", "*prestare a*") "possession" involving the *object* "ball"; the *relation change* (**R**-labelled) happens between the *donor* "Fred" and the *recipient* "John". As in the above example, most of the times Schank analysis

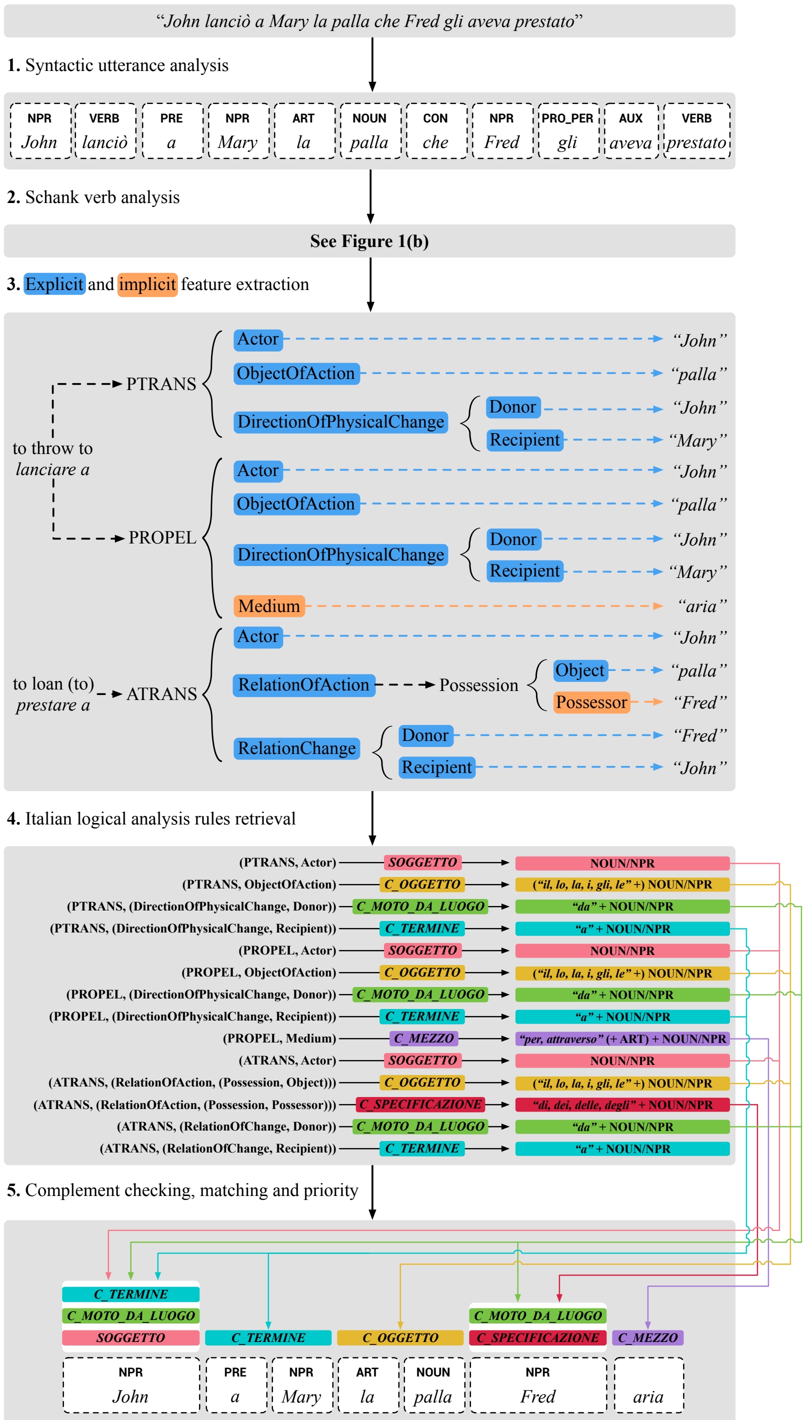


Fig. 2: PSRL phases: from sentence to logical analysis element extraction

discloses *contextual hidden information* involving the actions performed. In fact, while we expect that the PTRANS conceptualisation always comes with *explicit features* such as a donor (John), a recipient (Mary), and an object involving the action (a ball), we can also argue that probably the ball has been thrown through the air (a *medium*), as well as that the ball is Fred's, since it has been loaned by Fred to John (a *possession*). Such *implicit features* allow the same sentence chunk to assume different semantic facets at the same time (e.g., Fred is both the *explicit donor* and the *implicit possessor* of the ball), and may give more significance to its semantic content. Techniques of semantic role labelling for Italian such as logical complement analysis – where for instance, an object (*complemento oggetto*) or a complement regarding possession (*complemento di specificazione*) has typically more semantic meaning w.r.t. a complement involving times (*complemento di tempo*) or places (*complemento di moto a luogo*, *moto da luogo*, and so on) – could productively take advantage of this peculiar behaviour.

Driven by these considerations, PSRL consists of the following phases (applied to the above example in Figure 2):

1. **Syntactic utterance analysis** where the sentence is split into syntactic tokens with Italian NLP tools (e.g., OpenNLP tokeniser) and the syntactic information of each chunk is gathered from suitable dictionaries (e.g., MorphIt) or by means of Named Entity Recognition techniques (e.g., OpenNLP NER);
2. **Schank verb analysis** where all verbs in the sentence are isolated, and the complete Schank representation of the utterance is computed;
3. **Explicit and implicit feature extraction** where the system retrieves and populates all the features attached to each Schank action type involved (e.g., Actor, ObjectOfAction, DirectionOfPhysicalChange.Donor and DirectionOfPhysicalChange.Recipient as explicit features, and Medium implicit feature for PROPEL action type in Figure 2);
4. **Italian logical analysis rules retrieval** where each extracted feature is mapped to a logical complement of Italian language. The mapping is possible by querying a special repository containing two types of rules:
   - *feature-to-complement rules* of the type (SATN, FNT) → ICN, where (i) SATN is a Schank action type name, (ii) FNT is an iterative tree of features names of the type FN – i.e., a singleton feature name – or (FN, FNT) – required in cases where populated features are subfeature of other implicit or explicit features – and (iii) ICN is an Italian logical complement name. The intended meaning of a rule such as (ATRANS, (RelationOfAction, (Possession, Object))) → C_OGGETTO is that "the Object component of the Possession sub-feature of explicit RelationOfAction feature for a ATRANS action type is mapped to a *complemento oggetto*" (an object in the logical sense).
   - *complement-to-syntactic construction rules* of the type ICN → {SC}, where ICN is an Italian logical complement name and {SC} is a set (possibly a singleton[4]) of syntactic constructions to be matched in the question in order to recognise the complement content. As an example, the

[4] In Figure 2 all complement-to-syntactic construction rules have a singleton set in their right side not to overload the picture with too much information.

rule C_OGGETTO → (*il*, *lo*, *la*, *i*, *gli*, *le* +) NOUN/NPR means that "a noun or a named entity (possibly) preceded by either *il*, *lo*, *la*, *i*, *gli*, *le* particle is tagged as a *complemento oggetto*".

5. **Complement checking, matching and priority** where complements are extracted from the question. All information classified as explicit feature in Phase 3. is double-checked and formatted w.r.t. the set of SCs retrieved in Phase 4., and then tagged with all applicable ICNs. On the other hand, all information derived from implicit features extraction is added to the set of complements without additional checking, since such information is not available in the original question (e.g., "*aria*" as *complemento di mezzo* in Figure 2). Since sentence chunks may be tagged with several ICNs (e.g., the "*John*" NPR token), a unique ICN for each sentence chunk is chosen according to an Italian complement ranking list. For our first case study, QA^{2}SPR architecture applies the following precedence order among complements:
   (a) Subject (*SOGGETTO*);
   (b) Object (*C_OGGETTO*);
   (c) Complement regarding possession (*C_SPECIFICAZIONE*);
   (d) Complements regarding places (e.g., *C_MOTO_A_LUOGO*);
   (e) Complements regarding times (e.g., *C_TEMPO*);
   (f) Other complements.

   According to the list above, the "*John*" token is regarded as a *SOGGETTO*, and "*Fred*" token is a *C_SPECIFICAZIONE.*

The ICN choice based on a complement ranking list represents the first of three prioritisation levels allowed by PSRL. As already pointed out, also keyword retrieval phase (Subsection 3.1) and candidate answer passage ranking (Subsection 3.2) benefit from PSRL inner ranking mechanism.

3.1 PSRL for keyword retrieval

Complements extracted after PSRL phases are all good candidates as keywords for the subsequent document extraction phase. The simpler heuristic for keyword retrieval (as the one used in our first prototype) is to choose all devised complements as equally important keywords for document search without a precise order, but more complex combinations may be devised: a strict subset of (un)ordered complements – e.g., only (un)ordered subject and object – or even a keyword multi-search based on combinations of most important complements.

3.2 PSRL for candidate answer passages ranking

The third and last prioritisation phase of our technique is applied in candidate answer passages ranking. The preference order of Italian complement list may also induce a preference order among candidate answer passages extracted during the document processing phase. Consider the question "*Che animale é Pippo, l'amico di Topolino*?" – "What animal is Goofy, Mickey Mouse's best friend?". Suppose that the question classification phase correctly infers the expected answer type as *animal.* PSRL phases applied to such utterance extracts the set of complements shown in Figure 3(a). If all complements are used as keywords, the first two documents extracted during document processing phase – e.g., through Google IT Custom Search – are a Wikipedia page (`https://it.wikipedia.org/wiki/Pippo`), and a blog (`https://www.orgoglionerd.it/articles/2014/06/che-razza-di-a nimali-sono-personaggi-disney`).

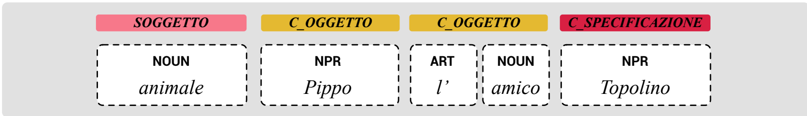
Thanks to PSRL analysis, the system is able to order paragraphs containing each single keyword based on the complement ranking induced by the list. Figure 3(b)–(c) shows the first two paragraphs QA^{2}SPR would actually retrieve for each keyword, and how such paragraphs would be ranked according to the complement priority list reported in Phase 5. of PSRL. During the subsequent answer processing phase, all substrings whose semantic content is compatible with the expected answer type *animal* are isolated from each paragraph (marked in Figure 3(b)–(c) with blue ellipses) with the aid of semantic resources such as MultiWordNet. In this case, the following answers are retrieved (in Italian alphabetical order): "*anatre*" ("ducks"), "*cane*" ("dog"), "*oche*" ("geese"), "*pantegana*" ("sewer rat"), "*papero*" ("gander"), "*Rattus Rattus*", "*topo*" ("mouse"), "*uccelli*" ("birds"). QA^{2}SPR is instructed to apply another layer of preference among answers (and retrieved documents in general) according to a ranked list of web page types: as an example, Wikipedia pages are regarded as containing more reliable information than blog or forum pages. As such, extracted answers are first ordered by web page types (first the Wikipedia page, then the blog page) and then ordered by paragraph ranking, obtaining the following answer order:[5] "cane", {"papero", "uccelli", "anatre", "oche", "Rattus Rattus"}, "cane", {"topo", "pantegana"}. In this example, the correct answer to the question is also the top ranked for PSRL.

4 QA^{2}SPR architecture

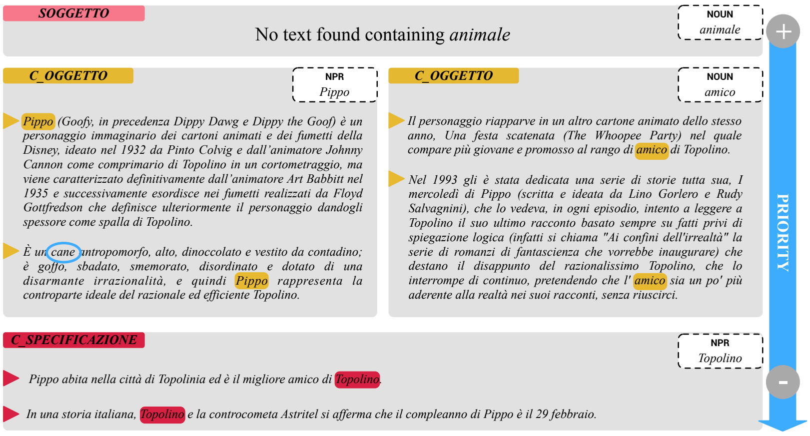
A complete diagram of QA^{2}SPR architecture customised for the Italian language – not reported here for space reasons – is freely available for download at `http://semantica.realt.it:81/QAASPR/KDWEB2017/Architecture.pdf`.

QA^{2}SPR conceptual design is slightly different from those of standard QASs, which usually consists of three basic modules: (i) a *question processing module*, whose main purpose is question classification; (ii) a *document processing module*, responsible of information retrieval; and (iii) an *answer processing module*, dedicated to answer extraction. In addition, two separated modules have been designed to manage documents coming from different web sources. The Knowledge Base Module (KBM) uses keywords to extract web data coming from knowledge base and annotated repositories (FreeBase, WikiBrain, DBPedia) – which makes it highly specialised for factoid answering, while the Full Open Domain Module (FODM) spans over the entire web to extract both structured (e.g., Wikipedia pages) and unstructured (e.g., free web text) information. Clearly, the FODM module is the one that depends the most on PSRL analysis, given that paragraph extraction and ranking phases are usually not required when information is extracted from knowledge bases.

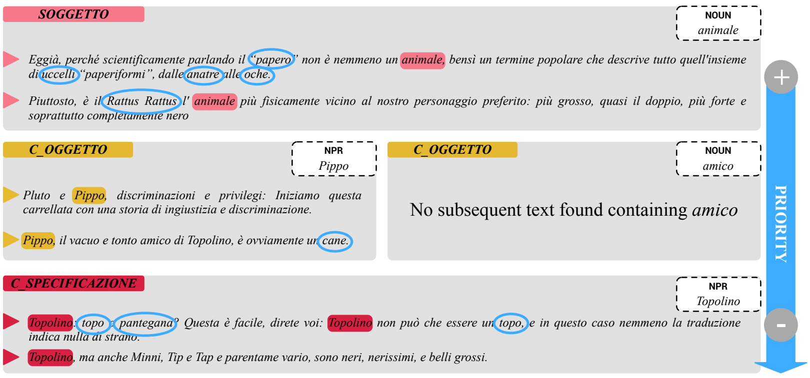
[5] Answers from (i) same paragraph, (ii) different paragraphs but for the same complement, and (iii) from different paragraph and different complement, but same complement type are equally prioritised and represented between square brackets.



(a)



(b)



(c)

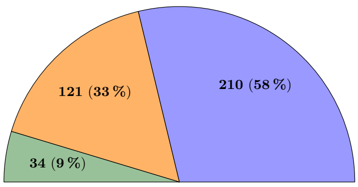
Fig. 3: (a) PSRL phases applied to the Goofy question; (b) Candidate passage ranking for Wikipedia web page; (c) Candidate passage ranking for blog web page.

5 Experimental results: a question a day ... for a year

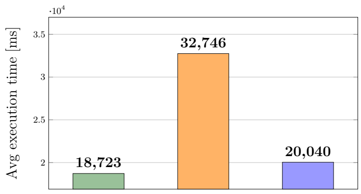
A working prototype of QA^{2}SPR architecture for the Italian language was developed in Java and hosted by a CentOS 6.8 Linux web server with four 64-bit cores running at 2.40 GHz and 4GB of RAM.[6] The system has been further asked

[6] The interested reader may contact the authors and access the online prototype.

to answer a set of 365 general knowledge questions randomly chosen from online repositories.[7] It is our intention in the near future to trial the prototype with standard question sets like the ones proposed in annual QAS tracks, e.g., the Text REtrieval Conference (TREC), CLEF workshops for European languages, and EVALITA tracks for NLP and speech tools evaluation for Italian. However, it has been noted that standard QAS track evaluation has remained somewhat controversial, since it is hard to classify the reliability of the answers to some question types (e.g., TREC and CLEF assessment as correct, unsupported, inexact, and incorrect) [19]. Despite CLEF and TREC ranking, each answer candidate the has been classified as (i) *correct* if at least the information required by the question is given. In such cases, answers have been further classified as *accurate* if they contain neither more nor less the information required, and *inaccurate* otherwise; (ii) *wrong* if they do not provide the required information; or (iii) *unavailable* if the system was not able to give a response (e.g., because no relevant document has been retrieved with the supplied keywords). In the remainder, an *available* answer trivially denotes either a wrong or a correct answer. We report a summary of overall results and performances in Figure 4(a)–(b), and individual scores related to KBM and FODM in Figure 5(a)–(f). The ratio of 58%



(a)



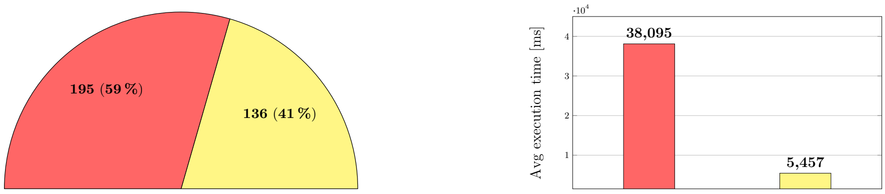
(b)

Fig. 4: QA^{2}SPR general performances: (a) Comparison among unavailable, wrong, and correct answers; (b) Comparison among execution times.

of correct answers – which increases to 64% if we ignore unavailable answers – represents in our opinion an encouraging push to follow the current research path, and suggests that even more satisfactory results might be achieved in view of future enhancements of the system. In this respect, Figures 5(c) and 5(e) clearly manifests where to focus our next efforts; in fact, answer extraction by KBM already exhibits excellent outcomes (80% of correct answers with a remarkable low average execution time), whilst more accurate PSRL heuristics for FODM are required (a correct answer a little over half the times). We remark, however, that parallel tests have been conducted showing the presence of the correct answer in at least one of the paragraphs extracted by FODM using PSRL 86% of the times.

Furthermore, the system shows a good work load division between answer retrieval by FODM and KB (59%-41% as revealed in Figure 5(a)), which confirms a proper choice of the test sample. The high variance between average execution times is clearly due to the different complexity carried by the two modules (e.g.,

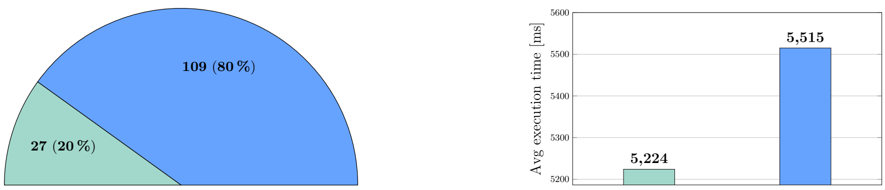
[7] The file `http://semantica.realt.it:81/QAASPR/KDWEB2017/Tests.pdf` with all the questions, answers, and execution times is freely available for download.



(a) (b)

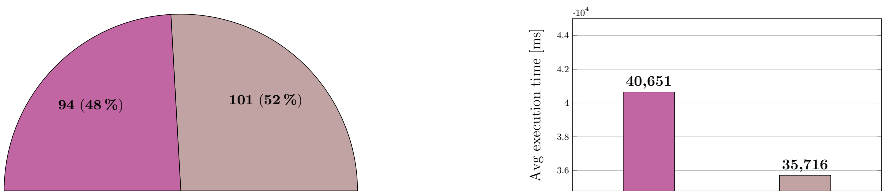
Fig. 5: QA^{2}SPR specific performances: (a) Comparison among available answers extracted with FODM and KBM; (b) Comparison among execution times.

number of sub-modules used, structured data from dataset vs. likely unstructured data from web document pages).



(c) (d)

Fig. 5: QA^{2}SPR specific performances: (c) Comparison among wrong and correct answers extracted with KBM; (d) Comparison among execution times.



(e) (f)

Fig. 5: QA^{2}SPR specific performances: (e) Comparison among wrong and correct answers extracted with FODM; (f) Comparison among execution times.

6 Conclusions and Future Work

We have already delineated in Section 5 some future investigation paths stemming from the present dissertation. In addition, we are currently considering feasible QA^{2}SPR applications in Ambient Semantic Computing (ASC). The main aim is to combine the semantic technologies offered by QA^{2}SPR architecture – such as NLP and ontology related research – with Ambient Intelligence (AI) and Ubiquitous Pervasive Computing (UPC) capabilities. In this regard, an exploratory study has been performed about the interactions between QA^{2}SPR and *MyElettra*, a system for real-time energy management and saving [2, 15]. The interaction among those systems already shows promising results, also thanks to an advanced methodology of ambient intelligence scheduling [3] and an improved mechanism to extract energy consumption best practices based on a default logic approach [4, 5].

References

1. Allam, A.M.N., Haggag, M.H.: The Question Answering Systems: a Survey. Int. J. of Res. and Rev. in Inf. Sci. (IJRRIS) 2(3) (2012)
2. Cristani, M., Karafili, E., Tomazzoli, C.: Energy Saving by Ambient Intelligence Techniques. In: Barolli, L., Xhafa, F., Takizawa, M., Enokido, T., Castiglione, A., Santis, A.D. (eds.) 17th Int. Conf. on Network-Based Inform. Sys., NBiS 2014, Salerno, Italy, September 10-12, 2014. pp. 157–164. IEEE Computer Society (2014)
3. Cristani, M., Karafili, E., Tomazzoli, C.: Improving Energy Saving Techniques by Ambient Intelligence Scheduling. In: Barolli, L., Takizawa, M., Xhafa, F., Enokido, T., Park, J.H. (eds.) 29th IEEE Int. Conf. on Advanced Inform. Netw. and Appl., AINA 2015, Gwangju, South Korea, March 24-27, 2015. pp. 324–331. IEEE Computer Society (2015)
4. Cristani, M., Olivieri, F., Tomazzoli, C.: Automatic Synthesis of Best Practices for Energy Consumptions. In: 10th Int. Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2016, Fukuoka, Japan, July 6-8, 2016. pp. 154–161. IEEE Computer Society (2016)
5. Cristani, M., Tomazzoli, C., Karafili, E., Olivieri, F.: Defeasible Reasoning about Electric Consumptions. In: Barolli, L., Takizawa, M., Enokido, T., Jara, A.J., Bocchi, Y. (eds.) 30th IEEE Int. Conf. on Adv. Inform. Netw. and Appl., AINA 2016, Crans-Montana, Switzerland, 23-25 March, 2016. pp. 885–892. IEEE Computer Society (2016)
6. Datla, V.V., Hasan, S.A., Liu, J., Benajiba, Y., Lee, K., Qadir, A., Prakash, A., Farri, O.: Open Domain Real-Time Question Answering Based on Semantic and Syntactic Question Similarity. In: Voorhees, E.M., Ellis, A. (eds.) Proc. of the 25th Text REtrieval Conf. (TREC), Gaithersburg, Maryland, USA. NIST (2016)
7. Fillmore, C.J.: Frame Semantics. Hanshin Publ. Co., Seoul, South Korea (1982)
8. Lenci, A., Johnson, M., Lapesa, G.: Building an Italian FrameNet through Semi-automatic Corpus Analysis. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proc. of Int. Conf. on Lang. Resour. and Eval. (LREC), Valletta, Malta. Eur. Lang. Resour. Assoc. (2010)
9. Loni, B.: A Survey of State-of-the-Art Methods on Question Classification. Literature Survey, Published on TU Delft Repository (2011)
10. Moldovan, D.I., Pasca, M., Harabagiu, S.M., Surdeanu, M.: Performance Issues and Error Analysis in an Open-Domain Question Answering System. ACM Trans. Inf. Syst. 21(2) (2003)
11. Montemagni, S., Barsotti, F., Battista, M., Calzolari, N., Corazzari, O., Lenci, A., Zampolli, A., Fanciulli, F., Massetani, M., Raffaelli, R., Basili, R., Pazienza, M.T., Saracino, D., Zanzotto, F., Mana, N., Pianesi, F., Delmonte, R.: Building the Italian Syntactic-Semantic Treebank. Springer Netherlands, Dordrecht (2003)
12. Pundge, A.M., S.A., K., Mahender, C.N.: Question Answering System, Approaches and Techniques: a Review. Int. J. of Comp. Appl. 141(3) (2016)

13. Md. Arafat Rahman, Md-Mizanur Rahoman: sJanta: An Open Domain Question Answering System. In: Kando, N., Joho, H., Kishida, K. (eds.) Proc. of the 11th Conf. on Eval. of Inf. Access Technol. (NTCIR), Natl. Cent. of Sci., Tokyo, Japan. Natl. Inst. of Inf. (NII) (2014)
14. Ray, S.K., Singh, S., Joshi, B.P.: A Semantic Approach for Question Classification using WordNet and Wikipedia. Pattern Recognit. Lett. 31(13) (2010)
15. Scannapieco, S., Tomazzoli, C.: Ubiquitous and Pervasive Computing for Real-Time Energy Management and Saving – A System Architecture. In: Barolli, L., Enokido, T. (eds.) Innov. Mob. and Internet Serv. in Ubiquitous Comput. (IMIS). Adv. in Intell. Sys. and Comput., vol. 612. Springer Int. Publ. AG (2017)
16. Schank, R.C.: The Fourteen Primitive Actions and Their Inferences. Tech. rep., Stanford Univ., Stanford, CA, USA (1973)
17. Shen, D., Lapata, M.: Using Semantic Roles to Improve Question Answering. In: Eisner, J. (ed.) Proc. of the Joint Conf. on Empirical Methods in NLP and Comput. NLL (EMNLP-CoNLL), Prague, Czech Republic. ACL (2007)
18. Tonelli, S., Pighin, D., Giuliano, C., Pianta, E.: Semi-Automatic Development of FrameNet for Italian (2009)
19. Wang, M.: A Survey of Answer Extraction Techniques in Factoid Question Answering. Tech. rep., Dep. of Comp. Sci., Univ. of Stanford (2006)
20. Ye, Z., Jia, Z., Yang, Y., Huang, J., Yin, H.: Research on Open Domain Question Answering System. In: Li, J., Ji, H., Zhao, D., Feng, Y. (eds.) Nat. Lang. Process. and Chin. Comput. (NLPCC), Nanchang, China. LNCS, vol. 9362. Springer (2015)