2009 Ninth International Conference on Intelligent Systems Design and Applications

An experiment about using copulative and comparative sentences as constraining relations

Andrés Soto
Computer Science Department
University of Carmen
Ave. 31, Ciudad del Carmen, CP 24160, Campeche, México.
email: soto_andres@yahoo.com

José A. Olivas, Francisco P. Romero, Jesús Serrano-Guerrero
Department of Information Systems and Technologies
University of Castilla La Mancha
Paseo de la Universidad 4, 13071-Ciudad Real, Spain.
email: JoseAngel.Olivas, FranciscoP.Romero, Jesus.Serrano@uclm.es

***Abstract*— Existing search engines and question-answering (QA) systems have made possible processing large volumes of textual information. Current work on QA has mainly focused on answering two basic types of questions: factoid and definition questions. However, the capability to synthesize an answer to a query by drawing on bodies of information which reside in various parts of the knowledge base is not among the capabilities of those systems. In this paper, a system oriented to infer query answers from a collection of propositions expressed in natural language is introduced. By means of a specific example, it is outlined how the system proceeds to face those situations. This approach is based on the use of formal constraining relations modeling copulative and comparative sentences. Combining those propositions with others contained in different knowledge bases and applying deduction rules, the desired answer could be obtained.**

Keywords- generalized constraints, natural language, prototypical forms.

I. INTRODUCTION

Existing search engines have many remarkable capabilities and their performance is improving constantly. Information Retrieval (IR) and Information Extraction (IE) systems have made possible the processing of large volumes of textual information. However, they present serious problems for answering specific questions formulated by users.

Information retrieval (IR) systems focus on searching relevant documents for general user queries. Their output is a ranked list of documents ordered by their similarity to the query. One of the main requirements for these systems is the ability to deal with huge amounts of textual information in very short times. This demand has imposed several restrictions on IR models, which are mainly based on statistical methods rather than on linguistic ones.

Recent research in QA has been mainly fostered by the TREC[1] and CLEF[2] conferences. Up to the moment, they have mainly considered only a very restricted version of the general QA problem [1]: simple questions which assume a definite answer typified by a named entity or noun phrase, such as factoid questions or definition questions, and exclude complex questions.

Deduction capability is the capability to synthesize an answer to a query by drawing on bodies of information which reside in various parts of the knowledge base. By definition, a QA system is a system which is expected to have deduction capability. However, deduction capability is not among the capabilities of those systems.

A prerequisite to mechanization of question-answering is precisiation of meaning of proposition drawn from natural languages (NL). The point of departure is the assumption that the meaning of a proposition in a NL may be represented as a generalized constraint. The basic idea proposed by Zadeh [2, 3] could be resumed as: 1) translate a NL proposition, NL(p), into a generalized constraint, GC(p), in order to obtain a precise version of it. Then, 2) to transform it by abstraction in a prototypical form, PtF(p), which could be used latter to deduce new information by deduction rules. The principal components needed are: (1) a dictionary from NL to Generalized Constraint Language (GCL); (2) a dictionary from GCL to Protoform Language (PfL); (3) a modular deduction database; and (4) a world knowledge database.

Generalized constraints GC are defined by Zadeh [3] as expressions of the form.

$$\text{GC: } X \text{ isr } R, \quad (1)$$

where X is the constrained variable; R is a constraining relation; and r is the modality of the constraint. The principal modalities of GCs are:

a) *Possibilistic (r = blank): X is R,*
b) *Probabilistic (r = p): X isp R,*
c) *Veristic (r = v): X isv R,*

where R plays the role of the corresponding distribution, possibility, probability or verity distribution according to each case. Some examples of GCs are shown in Table 1. Example 1 corresponds to a copulative sentence. Example 2 corresponds to a comparative sentence. And examples 3 and 4 correspond to sentences with active verbs.

[1] Text REtrieval Conference home page http://trec.nist.gov/
[2] Cross-Language Evaluation Forum home page http://www.clef-campaign.org/

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.224

Table 1 Examples of Generalized Constraints

Proposition p	Generalized Constraint GC(p)
Mary is tall	Mary. height is tall where Poss{X = u} = $\mu_{tall}(u)$.
Pat is taller than Mary	(Pat. height, Mary. height) is taller where Poss{ (X,Y) = (u,v) } = $\mu_{taller}(u,v)$.
Carol lives in a small city near San Francisco	Carol.Residence.Location is (small city near SF) where Poss{X = u} = $\mu_{small\ city\ near\ SF}(u)$.
Usually Robert returns from work at about 6 pm	(Time(Robert.returns.from.work) is about 6 pm} isp usually where Prob{X = u} = $\mu_{usually}(u)$.

Computation with Information Described in Natural Language, NL-Computation [4] involves a fusion of natural languages and computation with fuzzy variables, whose values are constrained. A fundamental thesis which underlies NL-Computation is that information may be interpreted as a generalized constraint. The key underlying idea is that information conveyed by a proposition may be represented as a generalized constraint.

SMILe (Soft Management of Internet e-Laboratory) group[3] has been working for several years developing tools for Information Retrieval (IR) and Question Answering (QA) systems using Soft Computing techniques. IRKA system is oriented to infer query answers from a collection of propositions expressed in natural language. The general structure of the system is outlined in this paper. At the same time, by means of a specific example, it is outlined how the system proceeds to infer an answer to a query from a collection of propositions retrieved from documents expressed in natural language.

Declarative sentences are by far the most common type of sentence. IRKA approach is to consider noun phrases and copulative, comparative and superlative sentences as constraints that describe characteristics and properties about the entities involved in those sentences. Combining those propositions with others contained in different knowledge bases and applying deduction rules, the desired answer could be obtained. Dictionaries relating those grammatical structures with formal relations expressing the previously mentioned constraints were defined in [5, 6, and 7].

II. Searching for not explicit information

Let's suppose that somebody wants to know who the second eldest of the Brontë sisters was. The Brontës were the world's most famous literary family. Charlotte, Emily and Anne Brontë were the authors of some of the best-loved books in the English language. The other three Brontë's children were Patrick Branwell, Maria and Elizabeth. Maria and Elizabeth were the oldest of all of them, while Patrick Branwell were the fourth, between Charlotte and Emily. Posting a query with the four words: second + eldest + Brontë + sister between double quotation marks by Google searcher returns no matches (see Tables 2 and 3), although thousands of pages are retrieved without them.

[3] http://smile.esi.uclm.es/

Table 2 Results by searching about the Brontë sisters

Query	References without ""	References with ""
Brontë sisters	686.000	288.000
Brontë sister	1.040.000	5.030
eldest Brontë sister	55.300	8
youngest Brontë sister	160.000	102
second eldest Brontë sister	6.460	0

Using the query "Who was the second eldest Brontë sister?" with several different QA systems did not improve much previous results (see Table 4). With Ask.com [8], the number of links retrieved is nearly the same as with Google search engine. In the other cases, the number of links is small but the user has to decide between several different and contradictory answers.

Current QA systems typically include a *question analysis* module to determine the type of question and to identify the expected answer type. In addition, this module is also intended to identify the question words that can be used to query the document collection in order to find relevant passages. Then, the *passage retrieval* module uses search engines to retrieve the set of passages (i.e. text paragraphs, sentences) more likely to contain the answer.

Subsequently a *filter* preselects small text fragments or sentences that contain strings of the same type as the expected answer. Finally, the *answer extraction* module analyze, considers the selected passages and the expected answer type as well as further statistical or syntactic aspects to locate the text strings that will represent the answer to the question.

Table 3 Results by searching about the Brontë daughters

Search for	Links obtained without ""	Links obtained with ""
Brontë daughter	461.000	662
Brontë daughters	336.000	190
eldest Brontë daughter	90.300	1
youngest Brontë daughter	365.000	0
second eldest Brontë daughter	15.200	0

Table 4 Results obtained using QA systems

QA system	Results
Answers.com [9]	2 snippets: 1) one about the *Brontë's family* and 2) *about Heath Ledger*
Semote [10]	10 snippets links about *different members of Brontë's family*
QuALim [11]	2 sentences: 1) one about Emily Brontë and 2) *one about Charlotte Brontë*
AnswerBus [12]	10 sentences about Emily Brontë.
DFKI QAS [13]	6 snippets with 3 possible answers: 1) Emily Brontë, 2) *Charlotte Brontë* and 3) *the Brontë Parsonage*
INFERRET [14]	5 different *cryptic* snippets about the Brontë's sisters: *Emily, Charlotte and Anne*.
OpenEphyra [15]	a link to *Brontë's web page* at Wikipedia
Ask.com [8]	4470 links about *Brontë's family* and many others

For example, in AnswerBus [16] after the extraction of answer candidate sentences, each sentence received a primary score. Then several techniques are used to refine the primary scores, including the determination of question type, use of a QA specific dictionary, named entities extraction, coreference resolution, and redundancy deletion. The final score that is used to determine the rank of an answer is a combination of the primary score and the influence of all the different factors.

In OpenEphyra system [17], a statistical framework estimates the probability of an individual answer candidate given a set of validation features that predict its relevance according to external resources, and a number of similarity features that exploit redundancy among the answer candidates. To estimate the relevance of an answer candidate, four external resources are used. Gazetteers and WordNet are used to check whether a candidate satisfies the relationship described in the question such as ISA(Shanghai, city) or IS-IN(Shanghai, China). The Web and Wikipedia are used in a data-driven approach. For instance, if there is a Wikipedia document whose title matches the answer candidate, the document is analyzed to obtain a tf-idf score, which is used as a relevance feature. Web snippets are used to calculate a word distance between an answer candidate and question keywords.

III. Constraining Formal Relations

Things around us could be described by sets of characteristics like color, size, form, flavor, etc. Adjectives, in noun phrases and copulative sentences, are used to describe the attributes of the entities. Each attribute has a value in certain dimension like age, height, length, etc. Adjectives usually have relative meanings, with many possible degrees in some dimension.

Copulative sentences are the most frequent in English. The copulative sentence "Mary is tall" could be expressed by the constraining formal relation [5, 6]:

'Mary' has_chrc height value tall degree m1. (2)

As long as Mary is assumed to be a person, the following relations could be deduced:

'Mary' isA person. (3)
tall_Mary isA tall_person. (4)
tall_person subClassOf person. (5)
tall_person has_attrib tall degree m. (6)

Dixon [18] distinguishes between two kinds of semantic opposition for adjectives:

Antonym: antonym pairs (i.e. large-small) do not provide absolute descriptions, but relative. They occur frequently in comparative constructions and establish a converse relation: if "A is longer than B", then "B is shorter than A".

antonym(PoleA, MeaningA, PoleB, MeaningB). (7)

Complement: in complement pairs, as married and single, the denial of one term implies the assertion of the other and vice versa. True complements cannot occur in comparative constructions; they give complete descriptions.

complement(PoleA,MeaningA,PoleB,Meaning). (8)

Antonym adjectives define a dimension for an attribute with two poles with infinite values between them. "Mary is tall" just means that Mary's height is close to the tall pole, although that other values as very tall and too tall will be closer to it. Each pole defines a fuzzy set with its membership function, i.e. μ_{tall} and μ_{short}. As a value is closer to tall pole, μ_{tall} increases and, at the same time, μ_{short} decreases; and vice versa.

Antonym adjectives define a finite and totally ordered label set according to the selected dimension as established by the ordinal fuzzy linguistic approach [19]. A mid term representing an assessment of "approximately 0.5" could be defined in order to keep odd the cardinality of the set. This way, the semantic of the linguistic term set is established from the ordered structure of the term set.

The comparative sentence "Mary is taller than Pat" could be expressed by the constraining formal relation:

C('Mary', is, more, tall, 'Pat', m1). (9)

where m1 is the membership degree. This sentence expresses that Mary is closer to the tall pole than Pat is. Therefore, Pat is closer to the opposite pole, the short one:

C('Pat', is, more, short, 'Mary', m2). (10)

where the degree of membership of Pat to the short pole is greater than Mary's.

The superlative degree of an adjective (or adverb) describes the relational value of an adjective (or adverb), comparing it to many or all others of its kind.

The superlative sentence "She is the most beautiful of her group" could be expressed by the constraining formal relation:

S('She',is,most,beautiful, "of her group", m3). (11)

IV. Example about the Brontë sisters

IRKA architecture is outlined in Fig. 1. First, the question is introduced by the Communication Interface and processed by the Question Processor, which transform it in a query for BUDI[4] meta-searcher [20]. At the same time, the original question is transformed into a goal to be proved while answering the question by the Inference Engine.

The documents retrieved by BUDI are processed by a Natural Language Processor tool composed by a scanner, a parser and an interpreter. The lexical analyzer [21] search the documents, collecting those sentences which contain terms included in the query in order to reduce the amount of information to process. As more terms from the query appear in a sentence, the relation between the sentence and

[4] BUsqueda DIfusa: Fuzzy Search in Spanish

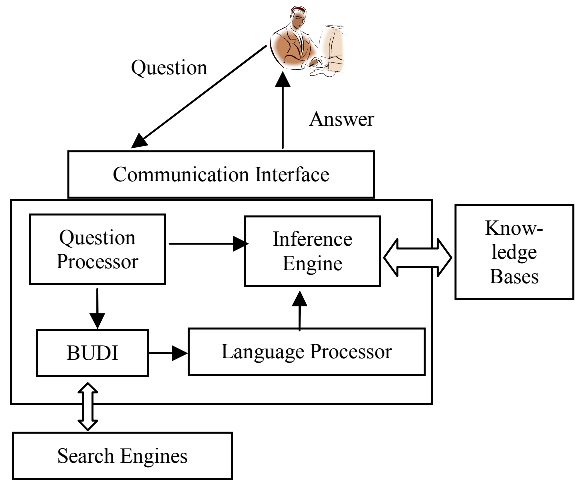

Fig. 1 IRKA architecture

the query is considered higher; thus it receives a higher score.

Equation 12 is used to filter retrieved sentences.

$$q \geq \lfloor (Q-1)^{0.5} \rfloor + 1 \qquad (12)$$

In this formula, q is the number of matching words in the sentence; and Q is the total number of matching words in the query. For example, if a query contains three words, then an answer candidate sentence should have at least two of them. When a sentence meets that condition, its score will be equal to the number of matching words it contains. Otherwise, it will be set to zero (0).

The sentences with score greater than zero are processed by a syntactic analyzer in SWI Prolog using DCG (Definite Clause Grammar) obtaining a parse tree for each one. By those parse trees, the different phrases and clauses contained in the sentences are identified, collected and processed by an interpreter, first independently and later altogether, transforming them into formal constraining relations according to [5, 6, and 7]. Later, those relations collected all together into a knowledge base are used to infer the query answer. Other knowledge bases and ontologies as WordNet and YAGO are also used by the inference engine.

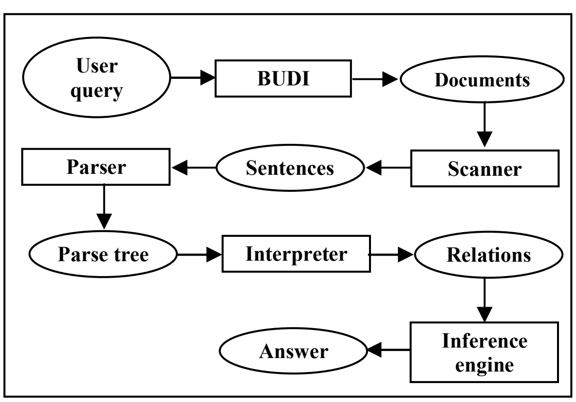

Fig. 2 General outline of IRKA flow process

Let's develop an example in detail in order to show the flow process of IRKA system. Let's suppose that the user query is: "who was the second eldest of the Brontë sisters?"

First, the main terms of the query, second + eldest + Brontë + sister (without using ""), are introduced to BUDI, which returns 24 links, the most suitable documents. All of them were links related with the Brontë's family: 12 about the family, 9 about Emily, 2 about Charlotte and 1 about Anne.

Those documents are then processed by the scanner which retrieves, in first place, the sentences including the four terms. One of those documents was Emily Brontë's page at Wikipedia, which includes the sentence:

"Emily was the second eldest of the three surviving Brontë sisters, being younger than Charlotte and older than Anne"

Then the selected sentences are parsed obtaining its corresponding syntactic tree representation showing the different phrases and clauses which conforms the sentence. Those phrases and clauses are then transformed into semantic relations by the interpreter, which are used latter to deduce the question answer. In this process, different ontologies are used.

The interpreter mainly transforms each phrase and clause into constraining relations. Noun phrases (NP) identify the entities related by the sentences and, therefore, by the resulting relations. Thus, most of the example concern with NPs.

Let's show this process by means of the example. The first part of the sentence, "Emily was the second eldest of the three surviving Brontë sisters", is a whole copulative sentence with two noun phrases: NP1) Emily and, NP2) the second eldest of the three surviving Brontë sisters. Analyzing further, two new NPs can be extracted: NP3) the second eldest and NP4) the three surviving Brontë sisters.

Analyzing NP4, a reference to a category could be recognized via the use of the plural form, sisters, according to Gasser [22]. Analyzing the noun phrase head, it is possible to recognize that 'Brontë' is a family name, and 'sisters' is a common noun. Therefore, the category or class identified by NP4 is made up of sisters, and will be identified using YAGO-NAGA [23, 24] categories:

Id1 means three_surviving_Brontë_sisters. (13)
Id1 subClassOf sisters (14)

Applying the same reasoning again and again, two new Ids and classes are obtained:

Id2 means surviving_Brontë_sisters. (15)
Id3 means Brontë_sisters. (16)
Id2 subClassOf sisters (17)
Id3 subClassOf sisters. (18)
Id1 subClassOf Id2. (19)
Id2 subClassOf Id3. (20)

Thus, using the categories suggested by Sowa [25], the following relations are obtained with the attributes: Brontë (their family name), surviving, and three, the number of members which belongs to the category.

Id3 has_attrib Brontë. (21)
Id3 has_attrib surviving. (22)
Id1 has_chrc cardinal_number value three. (23)

Looking at YAGO categories, it could be found that Brontë is a family name and Charlotte, Emily and Anne are given names.

Brontë isA family_name. (24)
Charlotte isA given_name. (25)
Emily isA given_name. (26)
Anne isA given_name. (27)

Therefore, it is possible to conclude that Emily Brontë is a person, which belongs to the Brontë family, another category defined in YAGO.

The sisters: Charlotte, Emily, and Anne are members of the category three_surviving_Brontë_sisters.

Charlotte isMemberOf Id1. (28)
Emily isMemberOf Id1. (29)
Anne isMemberOf Id1. (30)

Noun phrase 3 (NP3) refers to "the second eldest" of that category, the three surviving Brontë sisters. Therefore, there exists some Entity1 which is the eldest of the category:

S(Entity1, was, most, old, Id1, m1). (31)

If Entity1 is subtracted from the category Id1, then a new category appears three_surviving_Brontë_sisters2, which corresponds to category Id1 without Entity1.

send(Id1, substract([Entity1], Id4)). (32)
Id4 means three_surviving_Brontë_sisters2. (33)

Then, there should be some Entity2 which is the eldest of the new category.

superlative(Entity2, was, most, old, Id4, m2). (34)

And, Entity2 is the second eldest of the category identified by Id1. But, according to the whole copulative sentence, Emily was the second eldest of the category identified by Id1, thus she is Entity2.

Furthermore, analyzing the comparative part of the sentence: Emily…being younger than Charlotte and older than Anne", then two comparison relations are obtained:

comparative('Emily', was, more, young, 'Charlotte', m4).(35)
comparative('Emily', was, more, old, 'Anne', m5). (36)

According to propositions 28-30, Emily, Charlotte and Anne belong to the three surviving Brontë sisters. Thus, Charlotte should be Entity1 and Emily should be Entity2, the second eldest of the three surviving Brontë sisters. It should be observed that it refers to "surviving Brontë sisters", which is a subset (subClassOf) "Brontë sisters".

Repeating again the experiment but using 'Brontë daughters' in spite of 'Brontë sisters', the Brontë's family page at Wikipedia was retrieved. There, the following sentences were filtered:

> "In 1824 the four eldest Brontë daughters were enrolled as pupils at the Clergy Daughter's School at Cowan Bridge. The following year Maria and Elizabeth, the two eldest Brontë daughters, became ill, left the school and died; Charlotte and Emily were brought home."

The first sentence defines another class: the four eldest Brontë daughters, which is a subclass of daughters, which should be sisters each other.

Id5 means the_four_eldest_Brontë_daughters. (37)
Id6 means eldest_Brontë_daughters. (38)
Id7 means Brontë_daughters. (39)
Id5 subClassOf Id6. (40)
Id6 subClassOf Id7. (41)
Id7 subClassOf daughters. (42)
Id5 has_chrc cardinal_number four. (43)
X is SisterOf Y:- X isDaughterOf B, Y isDaughterOf B.(44)

The second sentence says that Maria and Elizabeth, the two eldest Brontë daughters, died, leaving alive Charlotte and Emily, two of the three surviving Brontë sisters. Thus, another class could be defined:

Id8 means two_eldest_Brontë_daughters. (45)
Id8 subClassOf eldest_Brontë_daughters. (46)

Applying relation 36, then

Id8 subClassOf Id6. (47)
Maria isA Id8. (48)
Elizabeth isA Id8. (49)

Therefore Maria and Elizabeth are daughters as well as sisters by clause 44. As long as they were the two eldest Brontë daughters, thus one of them should be the eldest and the other the second eldest daughter (and sister). Assuming it by order, Maria will be the eldest and Elizabeth the second eldest Brontë daughter.

S('Maria', was, most, old, Id8, m6). (50)

If 'Maria' is subtracted from the category,

send(Id8, substract(['Maria'], Id9)). (51)
Id9 means two_eldest_Brontë_daughters2 (52)

Then Elizabeth remains as the second eldest of the two eldest Brontë daughters, which is a subset (class) of the four eldest Brontë daughters.

S('Elizabeth', was, most, old, Id9, m7). (53)

Therefore, Elizabeth was the second eldest of the Brontë daughters (and sisters), while Emily was the second eldest of the three surviving sisters, the Brontë sisters that became famous and well known. As was mentioned before the Brontë family was composed by 5 daughters and a brother. The two eldest daughters, Maria and Elizabeth, died young;

therefore, they did not become famous and they are almost unknown.

The previous sentence does not say exactly that Maria, Elizabeth, Charlotte and Emily were the four eldest Brontë daughters but it looks normal to assume it, by context. Therefore

$$\text{Maria isA Id5.} \quad (45)$$
$$\text{Elizabeth isA Id5.} \quad (46)$$
$$\text{Charlotte isA Id5.} \quad (47)$$
$$\text{Emily isA Id5.} \quad (48)$$

From the previous propositions, the following one could be deduced:

$$\text{S(Maria, is, most, old, Id5, m8).} \quad (49)$$

If Maria is excluded from the set, a new set is obtained, where Elizabeth should be the eldest:

$$\text{send(Id5, substract([Maria], Id10)).} \quad (50)$$
$$\text{Id10 means four\_eldest\_Brontë\_daughters2.} \quad (51)$$
$$\text{S(Elizabeth, is, most, old, Id10, m9).} \quad (52)$$

By repeating that once more, a new set is obtained, made up by Charlotte and Emily.

Furthermore, knowing that Maria and Elizabeth died the following year,

$$\text{Maria diedOnDate following\_year.} \quad (53)$$
$$\text{Elizabeth diedOnDate following\_year.} \quad (54)$$

Therefore Charlotte and Emily became the two surviving daughters that were enrolled at Clergy Daughter's School. And Emily became the second eldest of the surviving sisters.

V. CONCLUSIONS

In this paper, the general structure of the IRKA system is introduced. By means of an example, the system process is outlined. Web documents retrieved by a metasearcher are filtered extracting the sentences which contains some of the query terms. Those sentences are parsed and latter converted into formal constraining relations. The algorithm to transform the phrases and clauses into constraining relations was explained in some detail by the example.

Combining those relations with others obtained from different knowledge bases, the inference engine derives the answer to the user question, using a fuzzy approach. There is a huge quantity of information contained in documents written in natural language, which could be retrieved and processed this way.

Although the current version of IRKA deals only with documents in English, it could also be applied to other languages as Spanish, assuming that the parse trees of the sentences could be obtained. In this case, the constraining relations could still be applied.

VI. ACKNOWLEDGMENT

This work was partially supported by TIN2007-67494 F-META project, MEC-FEDER, Spain and PEIC09-0196-3018 SCAIWEB-2 excellence project, JCCM, Spain.

VII. REFERENCES

[1] M. Montes Gomez, L. Villaseñor Pineda, and A. Lopez Lopez, “Mexican Experience in Spanish Question Answering Experiencia Mexicana en la Búsqueda de Respuestas en Español”, Computación y Sistemas, No. 001, julio 2008.

[2] L.A. Zadeh. “A note on web intelligence, world knowledge and fuzzy logic”. Data & Knowledge Engineering, 50, pp. 291–304, 2004.

[3] L.A. Zadeh. “From Search Engines to Question Answering Systems – The Problems of World Knowledge, Relevance, Deduction and Precisiation”. In Fuzzy Logic and The Semantic Web. Edited by Elie Sanchez, Elsevier B.V., 2006.

[4] L.A. Zadeh, “A New Frontier in Computation: Computation with Information Described in Natural Language”, IEEE International Conference on Information Reuse and Integration, 2007. IRI 2007.

[5] A. Soto, J.A. Olivas, and M.E. Prieto. “Using Generalized Constraints and Protoforms to deal with adjectives”. FUZZ-IEEE 2007 IEEE International Conference on Fuzzy Systems, Imperial College, London, UK, 2007.

[6] A. Soto, J.A. Olivas, and M.E. Prieto. “Using Generalized Constraints and Protoforms to deal with adverbs”, 5th Conference of the European Society for Fuzzy Logic and Technology EUSFLAT 2007, Ostrava, Czech Republic, 2007.

[7] A. Soto, J.A. Olivas, and M.E. Prieto. “Interpretando Información en Lenguaje Natural como Restricciones Generalizadas”, XIV Congreso Español sobre Tecnologías y Lógica Fuzzy ESTYLF 2008, Mieres-Langreo, pp. 625-630, 2008.

[8] Ask.com web page http://www.ask.com/.

[9] Answers.com web page http://www.answers.com/bb/.

[10] SEMOTE web page http://www.semote.com/.

[11] QuALim University of Edinburgh Question Answering system web page, http://demos.inf.ed.ac.uk:8080/qualim/.

[12] AnswerBus web page http://www.answerbus.com/index.shtml.

[13] DFKI Experimental Open Domain Web QA system web page http://experimental-quetal.dfki.de/.

[14] INFERRET web page, http://wikiferret.com/edw/pc/.

[15] OpenEphyra open source question answering system web page, http://sourceforge.net/projects/openephyra/.

[16] Z. Zheng. “AnswerBus Question Answering System”. Proceeding of HLT Human Language Technology Conference HLT 2002. San Diego, CA. March 24 - 27, 2002.

[17] N. Schlaefer, J. Ko, J. Betteridge, G. Sautter, M. Pathak, and E. Nyberg, “Semantic Extensions of the Ephyra QA System for TREC 2007”. In Proceedings of the Sixteenth Text REtrieval Conference TREC, 2007.

[18] R.W. Dixon. Where have all the adjectives gone? Mouton Publishers, 1982.

[19] F. Herrera, and E. Herrera-Viedma, “Aggregation operators for linguistic weighted information”. IEEE Transactions on Systems, Man, and Cybernetics, 27, pp. 646–656, 1997.

[20] J. Serrano-Guerrero, F.P. Romero, J.A. Olivas, J. de la Mata, A. Soto, “BUDI: Architecture for Fuzzy Search in Documental Repositories”, Mathware & Soft Computing 16 2009 pp. 71-85.

[21] C. Puente, J. A. Olivas. “Analysis, detection and classification of certain conditional sentences in text documents”. In Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU’08, pp. 1097-1104, Torremolinos, Spain, June 2008.

[22] M. Gasser. How Language Works. Edition 3.0, Indiana University, 2006.

[23] F.M. Suchanek, G. Kasneci, and G. Weikum. “YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia”, Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, pp. 697-706, 2007.

[24] G. Kasneci, F.M. Suchanek, M. Ramanath, and G. Weikum, "The YAGO-NAGA Approach to Knowledge Discovery", SIGMOD Record 2008.

[25] J.F. Sowa. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove, CA, 2000.