From Spoken Language to Ontology-Driven Dialogue Management

Dmitry Mouromtsev[1], Liubov Kovriguina[1], Yury Emelyanov[1], Dmitry Pavlov[1], and Alexander Shipilo[2]

[1] ISST Laboratory, ITMO University, Saint-Petersburg, Russia
www.ifmo.ru
{d.muromtsev,yury.emelyanov,lkovriguina,vergilius}@gmail.com
[2] Saint-Petersburg State University, Saint-Petersburg, Russia
www.spbu.ru
alexandershipilo@gmail.com

Abstract. The paper describes the architecture of the prototype of the spoken dialogue system combining deep natural language processing with an information state dialogue manager. The system assists technical support to the customers of the digital TV provider. Raw data are sent to the natural language processing engine which performs tokenization, morphological and syntactic analysis and anaphora resolution. Multimodal Interface Language (MMIL) is used for the sentence semantic representation. A separate module of the NLP engine converts Shallow MMIL representation into Deep MMIL representation by applying transformation rules to shallow syntactic structures and generating its paraphrases. Deep MMIL representation is the input of the module generating facts for the dialogue manager. Facts are extracted using the domain ontology. A fact itself is an RDF triple containing temporal information wrapped in the move type. Dialogue manager can accept unlimited number of facts and supports mixed initiative.

Keywords: spoken dialogue systems, domain ontology development, natural language processing, MMIL applications, paraphrase generation, information state approach

1 Introduction

The developed dialogue system is to assess customer support to the clients of digital television provider. The prototype of the system communicates with clients using chat window. The system is able to find out the problem and offer solution and give instructions how to fix the problem, otherwise it redirects the client to operator (human) or escalates the problem for another level of support.

The prototype deals with specially prepared textual data. Original training data are 150 human-human dialogues (5600 tokens) between users and technical support concerning troubles with digital TV subscription. The refinement included ellipsis recovery, discontinuity and spontaneous speech disfluency removal.

The main goal of the project was to develop a dialogue system which will classify the problem dynamically, will be able to accept unlimited number of facts and support mixed initiative. This determined choosing information state approach for dialogue management.

2 Related Work

There is a number of plan-based dialogue systems in healthcare[3] and technology[1] which support mixed initiative and interpret the dialogue using ontologies[1][2]. We decided to develop Natural Language Processing engine performing full and deep linguistic analysis instead of using n-gram or bag-of-words models. Shallow syntax transformation and paraphrasing rules applied to shallow syntax structures to produce invariant utterance structures have been also implemented. It is a well-known fact that direct mapping of the sentence to the system command is impossible excluding some very simple cases (e.g., greeting, acknowledgement).

3 Natural Language Processing Engine

NLP engine performs full linguistic analysis of the text in Russian language including tokenization, morphological analysis and syntactic analysis. SemSyn[3] parser tags the sentence, analyzes its syntactic structure and produces a dependency tree in the output, performing syntactic ambiguity resolution. SemSyn parser performs correct morphological analysis in 95% of the cases and syntactic analysis in 85-90% of the cases. Syntactic relations, ascribed by the parser, consider semantics of the word (or the construction): the rule invocation depends on the entries' characteristics in the Tuzov semantic dictionary[4] There are about 60 relations used by the parser to resolve prepositional ambiguity, distinguish subject and object, etc. These relations can be plainly mapped to semantic roles. The XML parse tree is sent to the semantic representation module which converts it to one or more MMIL components.

4 Sentence semantic representation

MMIL unit, a component, obligatory includes propositional content and dialogue type. Propositional content includes events and participants of the sentence.

Dialogue types proposed in MMIL can be mapped to speech acts (see full MMIL manual in [5]).

4.1 Utterance Semantic Representation in Shallow and Deep MMIL

MMIL generation module converts XML parse tree into a number of MMIL components. Each MMIL component is referred to a single sentence or parts of

[3] http://www.openclinical.org/dm_homey.html

the compound sentence, includes the following entities and their characteristics: type of the dialogue act, events (entities in time dimension), events characteristics (time, aspect, mode, person, voice, polarity), participants (entities not bounded in time dimension), participants characteristics (objType, mmilId, refStatus, number, person, gender, modifier), relations between entities (a special relation "propContent" links speech act type and proposition, relations between events and participants coincide with semantic roles).
MMIL generation module uses rules dealing with sentence semantics and semantics of grammar categories of tense, aspect, modality relying on the results of A.V.Bondarko functional grammar[6].

Deep MMIL is not specified by MMIL authors. In our work Deep MMIL representation is inspired by the ideas of generative grammar, “Meaning-Text” theory and its further applications. Semantic roles and grammar characteristics in Deep MMIL to a large extent correspond to the relations in deep sentence structure in the tradition of generative grammar. To produce Deep MMIL com-

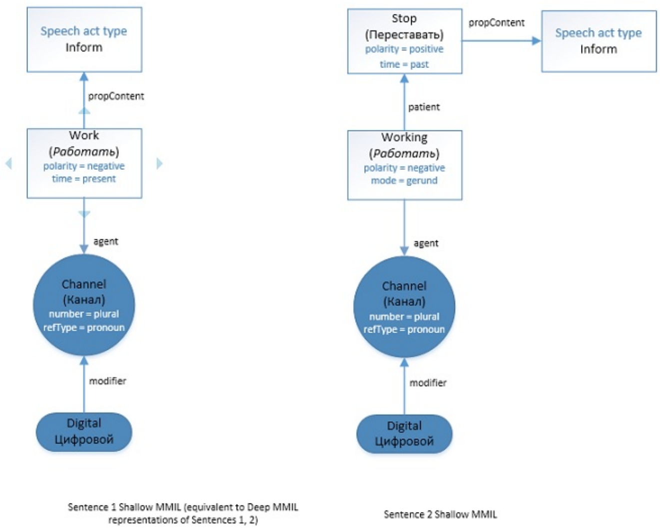

Fig. 1. Utterance representation in Shallow and Deep MMIL

ponent from Shallow MMIL component a number of transformations and paraphrases is performed. These modifications are aimed to produce the same deep representation for several shallow structures.

Shallow and Deep MMIL components for the input sentences "My digital channels do not work" and "My digital channels stopped working" can be seen in figure 1. Sentence 1 has the same representation for both Shallow and Deep MMIL.

5 Paraphrase Generation

Implementing paraphrase generation in a dialogue system seems necessary for several reasons. Firstly, the speaker can express the same meaning differently and the system should be able to handle it and react the same way. Secondly, paraphrases may be useful for the generation of clarification questions[7]. Moreover, it was shown that lemmatizing, synonym handling and paraphrasing improve performance of the dialogue system[8]. While developing the prototype of our dialogue system, we encountered that paraphrasing also contributes to minimization of ontology entities. There is a number of data-driven paraphrase generation techniques, which performance has already been evaluated[9]. A pivot method to generate paraphrases is to use statistical machine translation techniques when each utterance is translated into target language and then back into source[8]. In the cited paper authors used Google Translate API. In our project we used paraphrase generation method proposed by Apresyan and Cinman[10]. This method uses the notion of lexical functions introduced by I.A.Mel'čuk and A.K.Zholkovsky[11]
Mel'čuk defines[11] lexical function as following: "A lexical unit f is a function that associates with a given lexical unit [= LU] L, which is the argument, or keyword, of f, a set $\{L_i\}$ of (more or less) synonymous lexical expressions (the value of f) that are selected contingent on L to manifest the meaning corresponding to f:

$$f(L) = \{L_i\}. \tag{1}$$

Below are some examples of the lexical function OPER (ibid.) (do), (perform) [support verb]

- OPER 1 (strike N) = to be [on ~]
- OPER 1 (support N) = to lend [~]

In our study paraphrases have been manually extracted from the training corpus and matched to the set of lexical functions. SYN, OPER and ANTI turned out to be dominant lexical functions among the encountered in the corpus. The next subsection gives examples of ANTI paraphrasing rules according to the paper[10] and provides a mapping to the facts of the knowledge base using ANTI LF.

5.1 ANTI LF Paraphrase Rule

ANTI LF deals with antonyms and states that negative of the LU corresponds to the positive value of LU's antonym. Yu.D.Apresyan defines some antonyms as lexical units, for which the following statements are true: "P = !R" (ANTI2) and "stop R = begin !R" (ANTI1)[12, 18]. The following paraphrase rule is applied: X + Y => ANTI1(X) + ANTI2(Y). Consider the following example: "Two hours ago internet stopped working" ("Два часа назад перестал работать интернет"). It is clear that internet does not work but negation is hidden inside the verb "stopped". Shallow MMIL generated for this sentence will ascribe positive polarity to this verb meanwhile the fact sent to the dialogue manager must include false value of the property corresponding to the lexeme "to work". This paraphrase rule operates with polarity values in MMIL and as list of verbs denoting action beginning and termination. The given sentence will be paraphrased into MMIL intermediate structure corresponding to the sentence "Two hours ago internet began not to work" ("Два часа назад интернет начал не работать.") and to the final Deep MMIL representation where "work" will have the values of polarity = 'negative' and time = 'past'.

Paraphrase generation module uses specially compiled lexical resources, which specify the paraphrase rule, rule constraints and procedures changing Shallow MMIL structure. if all constraints are fulfilled corresponding paraphrase rule is invoked. It changes only the items of the MMIL component, no text is generated or affected. After all possible paraphrase rules have been applied Deep MMIL component is generated and sent to fact extraction module.

6 Domain Ontology

Firstly, we tried top-down approach to create domain ontology based on instructions provided for human customer support operator but in the end it proved to be unusable, because our concepts occurred rarely in real dialogues making impossible to build problem solving algorithms.

So, we switched to bottom-up approach and distinguished 6 types of problems with TV subscription and built the ontology according to empirical data.

Devices (TV set, router, set-top box, etc.), connectors (cable, switch, etc.) services and tariffs, user, etc. are represented as classes. Events are modeled as properties, like 'operational' property that has domain of Service and range of boolean, and restrict our modelling power. The problem was that utterances always has temporal reference which is difficult to model in OWL, so temporal level was introduced to the fact structure.

6.1 Lexical information in the domain ontology

Domain ontology incorporates the level of lexical semantics.

Synonyms are stored in listed as lemmas of the concept, i.e. for the boolean datatype property "insert" lemmas "insert", "stick in" and "put in" are stored. Some operations, expressed by properties like "insert", "switch on", "plug in" etc. have opposite operations. Each of these operations is stored in a separate subclass of the class defining the general name of these operations (see fig. 2[4]). Such representation of concepts that denote opposite operations allows to specify complex operations using part-whole relation, e.g. rebooting,a holonym, except lemma specifying can be defined with two its meronyms: "switch on" and "switch off".

[4] Visualized using http://www.ontodia.org/

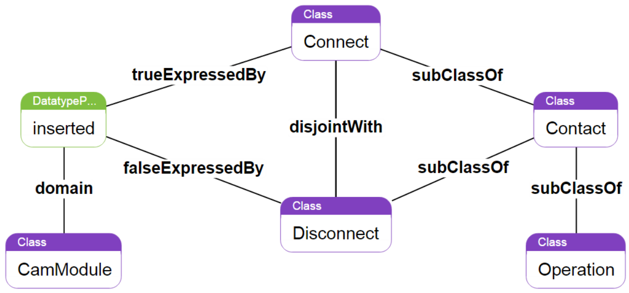

Fig. 2. Event representation in domain ontology

7 Extracting facts for the Dialogue Manager from Deep MMIL

Fact extraction is implemented in the separate model, having Deep MMIL in the output and generally producing an RDF triple wrapped in the fact type and move type tags.

7.1 Types of Facts in the Knowledge Base of the Dialogue Manager

The developed dialogue manager supports all main concepts of the information state approach. Dialogue manager's data structure pecularity is the following: each move, except the most simplest GREET and QUIT, includes factual information embedded in the "Fact" structure. There are three types of "Fact" in the system: "SimpleFact" (stores short answers), "Agreement" (stores boolean value for user's confirmation/rejection), "PropertyFact" (stores subject-predicate-value (for datatype properties) and subject-predicate-object (for object properties) triples, each corresponding to the elementary fact) and "AlternativeFact" (stores multiple PropertyFacts).

7.2 Fact Extraction Algorithm

This module is responsible for Deep MMIL parsing and extracting facts for the knowledge base which will be used for the dialogue manager. Basic idea of the algorithm is the following: each participant is the candidate subject (participant lemma is searched among lemmas of the domain ontology classes), each event is the candidate predicate (event lemma is searched among lemmas of the domain ontology properties)[13]. Combining move types, properties and temporal information allows to use the same ontology entities for different utterances: the only property tv:subscribed is stored in the ontology and corresponds to several situations, e.g. when the user 1) wants to subscribe a service, 2) has already subscribe it, 3) wants to unsubscribe the service. Extending speech act type and proposition distinction forth to the knowledge base and dialogue manager allows to reduce ontology size and keep rules more observable.

8 Dialogue Manager

8.1 Dialogue Move Engine

Dialogue move engine is responsible for maintaining context of dialogue, guiding basic flow of dialogue and question answering. It knows if user or system answered current question and maintains common facts for dialogue, as well as keeps plan for the next few actions. Dialogue move engine was created following the ideas described in GoDiS and implemented in TrindiKit with some modifications. We've added dialogue move type 'TELL' to provide informative messages to user and not to confuse this messages with answers to user questions. We've also added modification to update rules to allow accommodation of incoming facts when there are no relevant questions in QUD to implement mixed initiative dialog scheme. Data structures of information state holding moves, agenda, plan, common and private beliefs were created, along with update and select rules in Java programming language, unlike traditional implementation of GoDiS in TrindiKit in Prolog and therefore they use different abstractions for rule definition and application.

8.2 Domain Binding

GoDiS defines different functions on facts: relevant (if answer fact is relevant to question fact), resolves (if answer fact resolves question fact), combine (to combine question fact with answer fact to produce resolving fact). As stated above, we have information about concepts, properties and their types in ontology and use it doing relevant, resolves and combines functions on facts: to check data types, to use concept taxonomy trees to see if answer fits the question.

8.3 Problem solving

Dialogue move engine handles only basic part of dialogue flow. The more complex part of dialogue flow depends on things being said from both parties the more it depends on domain knowledge. The developed dialogue manager doesn't belong to command dialogue managers (like Smart Home control) or search systems or booking systems. The dialogue system should collaborate with user solving his problem, present him diagnostics options or asking to do specific task and tell if the main problem has gone. Simultaneously, user can ask questions about the data in information system, like balance, etc. The system may choose to redirect user to particular service to get additional help or query diagnostics system if there were any problems with that particular user. To implement this kind of system behaviour we chose to put the knowledge of interaction in rule system. Each time a fact comes from user, we check if any rule has fired. We used Drools Rule Engine[5] with custom domain-specific language layer.

[5] http://www.drools.org/

9 Evaluation, Conclusion and Future Work

Intermediate prototype evaluation performed on automatically generated Shallow MMIL components and its parts has shown the necessity of incomplete and contradictory facts handling. Future work implies massive research and development activities. NLP engine should be implemented with the parser analyzing spoken language syntax correctly. The algorithm of fact extraction needs refinement and elaboration. Ontology requires enlargement and the dialogue manager should be learnt to support partial and contradictory facts. Finally, handling large rule base is a hard task and another ways of storing and executing knowledge for problem-driven dialogue management should be elaborated.

10 Acknowledgements

This work was partially financially supported by the Government of the Russian Federation, Grant 074-U01.

References

1. Görz, G., Bücher, K., Ludwig, B., Schweinberger, F., Thabet, I.: Combining a lexical taxonomy with domain ontology in the erlangen dialogue system. In: Proceedings of the KI2003 Workshop on Reference Ontologies and Application Ontologies, Hamburg, Germany, September 16, 2003. (2003)
2. Maema, M.: OVR: A Novel Architecture for Voice-Based Applications. In: Thesis submitted for Master of Science. (2011)
3. Boyarsky, K., Kanevsky, E.: The semantic-and-syntactic parser SEMSIN. In: International conference on computational linguistics Dialog-2012. (2012)
4. Tuzov, V.: Semantic dictionary of the Russian language, http://emi.nw.ru/INDEX.html?0/Voc.html
5. Rojas-Barahona, L.M., Bazillon, T., Quignard, M., Lefevre, F.: Using MMIL for the high level semantic annotation of the French MEDIA dialogue corpus. In: Proceedings of the Ninth International Conference on Computational Semantics, Association for Computational Linguistics (2011) 375–379
6. Bondarko, A.V.: Functional grammar: a field approach. Volume 35., John Benjamins Publishing (1991)
7. Ebert, C., Lappin, S., Gregory, H., Nicolov, N.: Generating full paraphrases of fragments in a dialogue interpretation system. In: Proceedings of the Second SIGdial Workshop on Discourse and Dialogue-Volume 16, Association for Computational Linguistics (2001) 1–10
8. Gardent, C., Rojas-Barahona, L.M.: Using paraphrases and lexical semantics to improve the accuracy and the robustness of supervised models in situated dialogue systems. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP Seattle, Seattle, Washington, USA, meeting of SIGDAT, a Special Interest Group of the ACL. (2013) 808–813
9. Metzler, D., Hovy, E., Zhang, C.: An empirical evaluation of data-driven paraphrase generation techniques. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, Association for Computational Linguistics (2011) 546–551

10. Apresyan, Y., Zinman, L.: Perifrazirovanie na kompjutere" (in Russian). Volume 36., Vsesojuznij institut naucnoj b techniceskoj informacii (1998) 177
11. Mel'cuk, I.: Collocations and lexical functions. Volume 31., Clarendon Press, Oxford (1998) 23–53
12. Apresyan, Y.: Lexicheskaya semantika: Sinonimiceskije sredstva jazika (in russian), Nauka (1974) 367
13. Mouromtsev, D., Kozlov, F., Kovriguina, L., Parkhimovich, O.: A Combined Method for E-Learning Ontology Population based on NLP and User Activity Analysis. Volume 1254., CEUR-WS (2014)