NICE project (IST-2001-35293)

Natural Interactive Communication for Edutainment

NICE Deliverable D3.5-2a

English Natural Language Understanding Module

18 November 2004

Authors

Manish Mehta[1], Andrea Corradini[1], Niels Ole Bernsen[1]

1: NISLab, Odense, Denmark

Project ref. no.	IST-2001-35293
Project acronym	NICE
Deliverable status	Restricted
Contractual date of delivery	1 November 2004
Actual date of delivery	18 November 2004
Deliverable number	D3.5-2a
Deliverable title	English Natural Language Understanding Module
Nature	Report
Status & version	Final
Number of pages	13
WP contributing to the deliverable	WP3
WP / Task responsible	TeliaSonera
Editor	N/A
Author(s)	Manish Mehta, Andrea Corradini, Niels Ole Bernsen
EC Project Officer	Mats Ljungqvist
Keywords	ontology-based language processing, robust parsing, embodied conversational agents
Abstract (for dissemination)	This deliverable D3.5-2a from the IST/HLT NICE (Natural Interactive Communication for Edutainment) project describes NISLab's English natural language understanding module used in the second NICE Hans Christian Andersen system prototype.

1 Introduction

Speech recognition for children is a much harder problem than for adults due to age-dependent, acoustic and linguistic differences. This increases the complexity of understanding spoken language utterances, a task which is already made difficult by the presence of phenomena, such as false starts, ellipsis, incompleteness, word disorder and other kinds of ill-formedness. In such an environment, traditional parsers and grammars will not work well and attempts to perform complete syntactic analysis to account for all words in an utterance will break down. These problems have led many researchers to favour more semantic-driven approaches.

In our approach, we use a methodology where the first stage performs a shallow analysis of the input utterance by spotting the key phrases present. This is followed by a semantic/ syntactic analysis stage where the syntactic constraints are relaxed as needed, semantically meaningful parts of the input utterance are extracted, and domain-irrelevant parts are ignored. We believe that a purely semantic-driven approach will miss important cues provided by syntax whereas a hybrid approach helps us achieve the objective of accommodating spontaneous speech input and utilize syntactic knowledge as needed.

As we move towards systems for kids where the central character conducts face-to-face social interaction through verbal and non-verbal behaviours relating to the domains of expertise of the interlocutors, new research issues for language understanding architectures emerge, adding to the existing ones. The research challenges include 1) effectively dealing with ungrammatical and incomplete sentences, repetitions, repairs, and false starts which characterise spoken language communication; 2) dealing with children's speech that has different articulatory, grammatical and lexical features; 3) developing reusable ontological and lexical resources across characters and domains; 4) achieving a sufficient level of understanding of the social cues provided by the user during their interactions. The approach to dealing with these issues will be discussed in this document.

2 Natural Language Understanding Module

2.1 Components

The components of the NICE English Natural language understanding module (NLU) are:

- NLU Manager
- Keyphrase spotter
- Semantic analyser
- Concept Finder
- Domain spotter

2.2 Overall flow

In terms of general information flow, the NLU (Figure 2.1) receives the user utterance from the Speech Recognizer in terms of an N-best list (currently N = 3). The NLU analyses the top result from this list. The NLU Module Manager is responsible for communication across the different components in the module.

The Keyphrase spotter spots key phrases in the user utterance and converts them into syntactic/semantic categories. Thus, each set of key phrases is associated with syntactic/semantic categories.

The output of the Keyphrase spotter is passed on to the semantic analyser. The semantic analyser consists of a number spotter, a lexicon and a rule engine. The number spotter helps conversation on the user's and HCA's age. The lexicon entries consist of syntactic/semantic categories for individual words. After passing through the number spotter and lexicon, the processed user sequence is a sequence of semantic and syntactic categories. The rule engine processes this sequence by applying rules defined on the presence of certain semantic/ syntactic categories at specific positions in the user sequence. The resultant sequence is sent to the FSA processor. It acts as the deepest level of parsing. If the user sequence is able to traverse an FSA, the result corresponding to that FSA is the output semantic representation from the Semantic analyser. The FSAs are developed offline from training corpora.

At the next stage of processing, the Concept finder provides a mapping between semantic representations and a domain-oriented ontological representation in terms of concepts.

The Domain spotter finds the domain by mapping the concepts to their respective domains. This mapping is defined at design-time.

The final output consisting of concept(s)/subconcept(s), property, dialog act and domain is sent to the Character module via the Input fusion module.

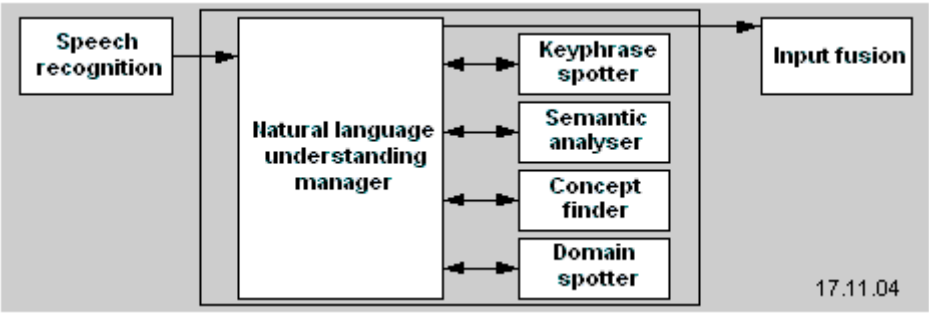

Figure 2.1. Natural language understanding module architecture.

2.3 Keyphrase spotter

This is the first component of the NLU and it is implemented in C++. The keyphrases are organized in terms of domains. Figure 2.2 shows the resources of the Keyphrase spotter. The Keyphrase spotter spots keyphrases in the user utterance and replaces them with their respective semantic/syntactic category. The Keyphrase spotter has a set of words labelled with semantic and syntactic tags, which are also used in the sentence-level parsing.

The key phrases have been found through analyses of real conversational data and armchair corpora which we have collected/created. This stage performs a shallow level of analysis, thereby ensuring that when faced with misrecognized utterances, the key phrases that are domain-related are extracted, and a wider acceptance of utterances than a sentence grammar could manage is achieved. As spoken language utterances are casual and full of ungrammatical phenomena, focusing on the keyphrases-only can still enable partial or full understanding of the user's intention. The output of the keyphrase spotter is of three types: single words, and syntactic and semantic categories.

Let us look at some examples to get a better idea of the processing of the Keyphrase spotter.

User Utterance: I like many of your **fairy tales.**
Keyphrase Module Output : I like many of your <fairytale:general>

User Utterance: I like **ugly duckling** and **princess and peas.**
Keyphrase Module Output : I like <fairytale:Ugly_Duckling> <fairytale:princess_peas>

User Utterance: **Ugly duckling** is **dear to my heart**
Keyphrase Module Output : <fairytale:Ugly_Duckling> <verb:like>

User Utterance: Did your father use to **repair shoes**
Keyphrase Module Output : Did your father use to <profession:shoemaker>

User Utterance: Did your mother use to **wash clothes**
Keyphrase Module Output : Did your mother use to <profession:washerwoman>

User Utterance: what did your father do to **earn a living**
Keyphrase Module Output : what did your father do to <profession:general>

User Utterance: Your fairy tales are dear to my heart
Keyphrase Module Output : Your <fairytale:general> are <verb:like>

2.4 Semantic analyser

The Semantic analyser has three components.

1. Number Spotter
2. Lexicon
3. Rule Engine
4. FSA Processor

2.4.1 Number Spotter

The Number Spotter is used to spot numbers in user input. Let us look at some examples.

Example: I am **fifteen** years old.
I am <number:fifteen> years old.

Example: You must be hundred years old.
You must be <number:100> years old.

Example: Were you born in eighteen sixty five.
Were you born in <number:1865>.

2.4.2 Lexicon

At the next stage, syntactic and semantic categories for single words are retrieved from a lexicon. The lexicon represents linguistic objects and semantic hierarchies. A lexical entry contains knowledge about each word in terms of its syntactic and semantic category. Words that occur in more than one semantic category have their actual meaning detected through rules as described in the next section. Inflected forms of verbs and morphological variants of a word, such as plurals of a noun, are provided with a separate entry. Relationships of synonymy are encoded by having the same semantic category for the involved words. These synonyms are near-synonyms in the strictest sense. Semantic relationships are encoded through a parent-child relationship in the lexicon. This semantic categorization allows for inheriting the properties of the categories and subcategories of which an entity is a member.

To clarify the concepts introduced so far, let us look at some examples.

User Utterance: Your fairy tales are dear to my heart.
Keyphrase Module Output: Your <fairytale:general> are <verb:like>.
After number spotter : Your <fairytale:general> are <verb:like>
After taking entries from the lexicon: <HCA> <fairytale:general> <aux:are> <verb:like>.

User Utterance : Did your father use to repair shoes
Keyphrase Module Output : Did your father use to <profession:shoemaker>.
After number spotter : Did your father use to <profession:shoemaker>.
After taking entries from the lexicon: <aux:did> <hca> <family:father> **use to** <profession:shoemaker>.

2.4.3 Rule Engine

Word sense disambiguation is a crucial component of a natural language understanding module. It represents the part of the human language understanding process which finds the appropriate meaning for an ambiguous word within a sentence from a range of possibilities. The next stage of analysis involves choosing appropriate meaning representations for a range of possibilities. A grammar in which terminal/non-terminal symbols are all semantically meaningful constituents is defined. The semantic and syntactic categories are used to write the grammar rules. The hierarchical representation of lexical knowledge through the lexicon helps in rule-based inferencing. This stage also involves detecting the speech acts listed in Table 2.1. Conversational agents which operate in a social context have to understand the

importance of thanks, or user praise, and to acknowledge them. The table also shows the representation of social cues in user utterance in terms of speech acts.

Dialogue act	Dialogue act type	Explanation
User_Opinion	Positive	when user agrees with something (e.g. “yeah sure I like ugly duckling”)
User_Opinion	Negative	when user disagrees with something, e.g. “No I don’t want to talk about it”
User_Opinion	General	when user has opinion about something ,whether the user likes, dislikes, praises, etc. something is indicated by the property
User_Opinion	Thanks	when user expresses gratitude for something
User_Opinion	Insulting	user says socially inappropriate words or phrases (e.g. “none of your business,” “are you dumb”)
User_Opinion	praise	user praises something (e.g. “that is great”, “it is wonderful”)
Question	Location	when the user asks a question about a particular location (e.g. “where did you write ugly duckling”, “which place were you born”)
Question	Reason	when the user asks a question asking the reason behind it (e.g. “why did you leave copenhagen”, “how come you like ugly duckling”)
Question	Yes/no	when the user asks a yes/no type question (e.g. “do you like ugly duckling”, “is this your study”)
Question	Person	when the user asks a question about a person, e.g. “who was your father”
Question	general	a general question about a particular thing, e.g., what did ...
Question	Time	a question asking about the time when something happened (e.g. “when did you write ugly duckling”)
Request	Listen	user wants to hear about something (e.g. “could you tell me ...”, “tell something about…, “I want to hear about ...”)
Request	Tell	user wants to tell the character something (e.g. “I want to tell you ...”, “can I tell you ...”)
Meta	Repetition	user asks to repeat something, e.g. “could you please repeat”
Meta	clarification	user clarifies what he has said
Meta	correction	user corrects HCA
greeting	Ending	user says goodbye to a character, (e.g. see you later, “catch you later”)
greeting	beginning	user says hello

Table 2.1. Dialogue acts.

The tussle for robust semantic interpretations results in an entire disregard for syntactic constructs and knowledge but it is important in many situations that syntactic nuances are recognized. Tense plays a key role in distinguishing between “It was a pleasure meeting you” and “it is a pleasure meeting you” where in the first utterance the user bids goodbye and in the former the user praises seeing HCA. Syntactic knowledge also helps in deducing the question speech acts. Sentences with declarative, imperative yes/no question also follow a certain grammatical structure that is used in making rules based on the syntactic categories.

2.4.4 FSA Processor

Two components are used for processing of Finite State Automatons. FSA processing represents the deepest level of parsing. It helps extract meaning out of sentences where the sequence by itself does not make any sense. There are two components involved in the development of FSAs.

1. FSA Compiler: used to build FSAs from the training corpus. This process is done offline.
2. FSA Processor: during run-time, the FSAs are read by the FSA processor. If the user sequence is able to traverse an FSA, the result corresponding to that FSA is the output semantics from the NLU.

Let us look at an example to better understand the processing.

Example : What is the big deal about your life.
After Semantic Analysis: <Question:general> <big> <deal> <hca> <life> does not make much sense.
The FSA processor goes through the FSAs and tries to see which FSA this sequence traverses.
Result after FSA processing : <Question:general> <hca_who>

2.5 Concept Finder

The next stage of processing provides a mapping between semantic representations and a domain-oriented ontological representation. The organization of knowledge has to be related to linguistic system levels of organization such as grammar and lexis. A non-linguistic ontology would be limited when used in applications for language processing. Domain ontology captures knowledge of a particular domain and serves as a more direct representation of the world. Ontological relationships 'is-a' and 'a-kind-of' have a lexical counterpart in hyponymy and likewise subsumption has hypernymy [1]. The part-whole relationships meronymy and holonomy also form hierarchies. The relationship parallelism suggests that lexical relationships and ontology are the same but a semantic hierarchy might serve as a basis for a useful ontology and serve as the basis of grounding of semantic representation and can at most be called an ersatz ontology [2]. Different representations bridge the gap from linguistic inputs to the kind of non-linguistic knowledge needed in order to perform a variety of tasks involving the meaning of linguistic inputs.

Lets look again at some examples to better understand the processing.

Example: I like your room
After Keyphrase Module: I like your room
Lexicon output: <user> <verb:like> <hca> <room>
Semantic analyser Output: <user_Opinion:general> <verb:like> <study:general>
Concept Finder Output: <dialogue_act:user_opinion> <dialogue_act_type:general> <property:like> <location:study>

Example : Did you enjoy when you were a kid
After Keyphrase Module: Did you enjoy <time:childhood>
Lexicon output: <aux:did> <hca> <verb:like> <time:childhood>
Semantic analyser Output: <Question:general> <verb:like> <time:childhood>
Concept Finder Output: <dialogue_act:question> <dialogue_act_type:general> <property:like> <lifetime:childhood>

2.6 Ontological representation

Conceptualizing a domain helps in developing a clear understanding and appropriate reasoning for the domain. Research on ontology reuse is becoming increasingly widespread in the computer science community. Knowledge organization that is relevant across domains is recognized as being important and reusable ontological resources are sought after. To ensure the usability of an ontology, it should describe the domain knowledge in a generic way and provide an agreed understanding of the domain. Reusable ontology fragments can result in a great deal of time, effort and cost-savings, by avoiding to have to build from scratch. Many aspects of spoken dialogue interaction are common across domains. One of the clear uses of ontological reuse in a conversational system built for historical characters is to craft the concepts for his life in a generic way. A character's lifetime which comprises his birth, childhood, youth, adulthood, and old age, and his family which comprises his father, mother, brother, sister, wife, grandfather, and grandmother, represents general concepts about life which can be shared. We present below some of the concepts and/or sub-concepts that are used for the life domain as well as some of the properties which are shared across domains.

2.7 Common Properties shared across domains

- Number
    - few
    - lot
    - all
    - cardinals
- quality
    - good
    - bad
- emotion
    - funny
    - sad
    - scary
    - happy
- cognitive
    - know
    - remember
    - forgot
- like
- dislike
- leave
- visit
- live

2.8 Concepts and sub concepts for life domain

- Location
    - School
    - Neighbourhood

    - House
    - Study
    - Apartment
    - Birthplace
    - Hometown
    - Room
- Family
    - parents
    - mother
    - father
    - children
    - grandfather
    - grandmother
    - wife
    - general
    - brother
    - sister
    - children
    - sibling
    - grandchildren
    - son
    - daughter
- Lifetime
    - General
    - Birth
    - Childhood
    - Youth
    - Adult
    - old
    - Death

2.9 Domain spotter

Domain Spotting is done based on the output of the Concept finder phase by looking at each individual concept and property. The domain spotter maps these categories to their respective domains. It does concept/property-to-domain mapping for each of the domains.

Possible Domains for HCA are

1. Works: related to his fairytales, fairytale characters, novels, etc.
2. Life: related to his family, childhood, travels, etc.
3. Physical Presence: related to objects in his study, his physical appearance, etc.
4. User: user age, name, gender, nationality.
5. Meta: related to when the user does not understand something, asks HCA to repeat, corrects what HCA understood, etc.
6. Gatekeeper: related to HCA's role in the fairytale world with other characters.

3 References

[1] Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Daniel Jurafsky and James H. Martin. Prentice-Hall, 2000.

[2] Graeme Hirst: Ontology and the Lexicon. Handbook on ontologies 2004, 209-230.