Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 157–158.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

The Fyntour Multilingual Weather and Sea Dialogue System

Eckhard Bick
University of Southern Denmark
Odense
echard.bick@mail.dk

Jens Ahlmann Hansen
University of Southern Denmark
Odense
ahlmann@voicetech.dk

1 Introduction

The Fyntour multilingual weather and sea dialogue system provides pervasive access to weather, wind and water conditions for domestic and international tourists who come to fish for seatrout along the coasts of the Danish island of Funen. Callers access information about high and low waters, wind direction etc. via spoken dialogues in Danish, English or German. We describe the solutions we have implemented to deal with number format data in a multi-language environment. We also show how the translation of free text 24-hour forecasts from Danish to English is handled through a newly developed machine translation system. In contrast with most current, statistically-based MT systems, we make use of a rule-based apporach, exploiting a full parser and context-senstitive lexical transfer rules, as well as target language generation and movement rules.

2 Number Format Data

The Fyntour system provides information in Danish, English and German. A substantial amount of data is received and handled in an interlingua format, i.e. data showing wind speed (in m/s) and precipitation (in mm) are language-neutral numbers which are simply converted into language-specific pronunciations by specifying the locale of the speech synthesis in the VoiceXML , e.g.

<prompt xml:lang="da-DK"> 1 </prompt> ”en”
<prompt xml:lang="de-DE"> 1 </prompt> ”ein”
<prompt xml:lang="en-GB"> 1 </prompt> ”one”

In Germany, wind speed is normally measured using the Beaufort scale (vs. the Danish m/s norm), while visitors from English speaking countries are accustomed to the 12-hour clock (vs. the continental European 24-hour clock). These cultural preferences can be catered for by straightforward conversions of the shared number format data – performed by the application logic generating the dynamic VXML output of the individual languages.

However, the translation of dynamic data in a free text format, from Danish to English and Danish to German, – such as the above-mentioned forecasts, written in Danish by different meteorologists – is more complex. In the Fyntour system, the Danish-English translation problem has been solved by a newly developed machine translation (MT) system. The Constraint Grammar based MT-system, which is rule-based as opposed to most existing, probabilistic systems, is introduced below.

3 CG-based MT System

The Danish-English MT module, Dan2eng, is a robust system with a broad-coverage lexicon and grammar, which in principle will translate unrestricted Danish text or transcribed speech without strict limitations to genre, topic or style. However, a small benchmark corpus of weather forecasts was used to tune the system to this domain and to avoid lexical or structural translation gaps, especially concerning time and measure expressions, as well as certain geographical references and names.

Methodologically, the system is rule-based rather than statistical and uses a lexical transfer approach with a strong emphasis on source language (SL) analysis, provided by a pre-existing Constraint Grammar (CG) parser for Danish, DanGram (Bick 2001). Contextual rules are used at 5 levels:

1. CG rules handling morphological disambiguation and the mapping of syntactic func-

tions for Danish (approximately 6.000 rules)

2. Dependency rules establishing syntactic-semantic links between words or multi-word expressions (220 rules)
3. Lexical transfer rules selecting translation equivalents depending on grammatical categories, dependencies and other structural context (16.540 rules)
4. Generation rules for inflexion, verb chains, compounding etc. (about 700 rules)
5. Syntactic movement rules turning Danish into English word order and handling sub-clauses, negations, questions etc. (65 rules)

At all levels, CG rules may be exploited to add or alter grammatical tags that will trigger or facilitate other types of rules.

As an example, let us have a look at the translation spectrum of the weatherwise tedious, but linguistically interesting, Danish verb *at regne (to rain),* which has many other, non-meteorological, meanings *(calculate, consider, expect, convert ...)* as well. Rather than ignoring such ambiguity and build a narrow weather forecast MT system or, on the other hand, strive to make an “AI” module *understand* these meanings in terms of world knowledge, Dan2eng chooses a pragmatic middle ground where grammatical tags and grammatical context are used as *differentiators* for possible translation equivalents, staying close to the (robust) SL analysis. Thus, the translation *rain (a)* is chosen if a daughter/dependent (D) exists with the function of situative/formal subject (@S-SUBJ), while most other meanings ask for a human subject. As a default[1] translation for the latter *calculate (f)* is chosen, but the presence of other dependents (objects or particles) may trigger other translations. *regne med (c-e),* for instance, will mean *include,* if *med* has been identified as an adverb, while the preposition *med* triggers the translations *count on* for human “granddaughter” dependents (GD = <H>), and *expect* otherwise. Note that the *include* translation also could have been conditioned by the presence of an object (D = @ACC), but would then have to be differentiated from (b), *regne for (‘consider’).*

regne_V[2]
(a) D=(@S-SUBJ) :rain;
(b) D=(<H> @ACC) D=("for" PRP)_nil :consider;
(c) D=("med" PRP)_on GD=(<H>) :count;
(d) D=("med" PRP)_nil :expect;
(e) D=(@ACC) D=("med" ADV)_nil :include;
(f) D=(<H> @SUBJ) D?=("på")_nil :calculate;

It must be stressed that the use of grammatical relations as translation differentiators is very different from a simple memory based approach, where chains of words are matched from parallel corpora. First, the latter approach - at least in its

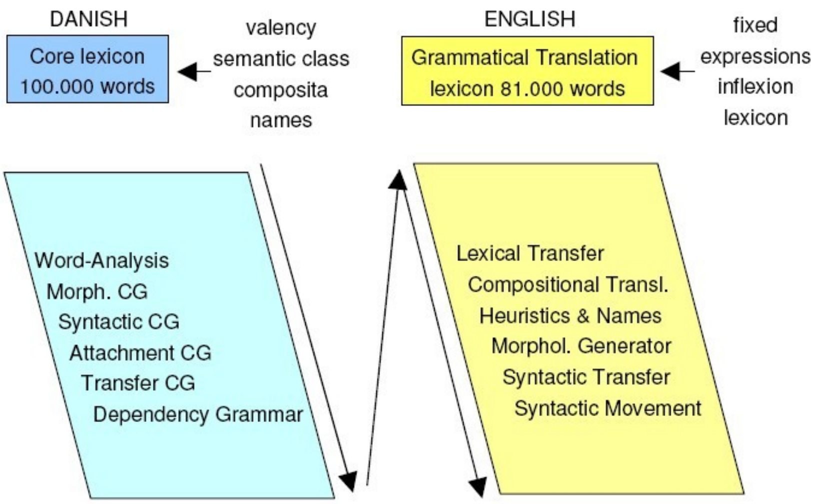


Fig 1: The Dan2eng system

naïve, lexicon-free version - cannot generalize over semantic prototypes (e.g. <H> for human) or syntactic functions, conjuring up the problem of sparse data. Second, simple collocation, or co-occurrence, is much less robust than functional dependency relations that will allow interfering material such as modifiers or sub-clauses, as well as inflexional or lexical variation.

For more details on the Dan2eng MT system, see http://beta.visl.sdu.dk/ (demo, documentation, NLP papers).

[1] The ordering of differentiator-translation pairs is important - defaults, with fewer restrictions, have to come last. For the numerical value of a given translation, 1/rank is used.

[2] The full list of differentiators for this verb contains 13 cases, including several prepositional complements not included here *(regne efter, blandt, fra, om, sammen, ud, fejl ...)*

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 159–160.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Dialog OS: an extensible platform for teaching spoken dialogue systems

Daniel Bobbert
CLT Sprachtechnologie GmbH
Science Park Saar
66123 Saarbrücken, Germany
bobbert@clt-st.de

Magdalena Wolska
Computational Linguistics
Universität des Saarlandes
66041 Saarbrücken, Germany
magda@coli.uni-sb.de

1 Introduction

With the area of spoken dialogue systems rapidly developing, educational resources for teaching basic concepts of dialogue systems design in Language Technology and Computational Linguistics courses are becoming of growing importance. Dialog OS[1] is an extensible platform for developing (spoken) dialogue systems that is intended, among others, as an educational tool.[2] It allows students to quickly grasp the main ideas of finite-state-based modelling and to develop relatively complex applications with flexible dialogue strategies. Thanks to Dialog OS' intuitive interface and extensibility, system implementation tasks can be distributed among non-technically- and technically-oriented students making the tool suitable for a variety of courses with participants of different backgrounds and interests. Below, we give a brief overview of the framework and outline some of the student projects in which it was used as a basis for dialogue management and modelling.

2 Dialog OS: a brief overview

Dialog OS is an extensible platform for managing and modelling (spoken) dialogue systems. It comprises an intuitive Graphical User Interface (GUI), default dialogue components, and a communications API to build new components. Dialog OS is written in Java and operates in a client-server mode. The central component can handle connections with an arbitrary number of client components (or "Devices", in Dialog OS terminology) via TCP/IP sockets. Technical requirements for Dialog OS are: 1 GHz Pentium, 512 MB RAM, Windows 2000/XP, Java Runtime 1.5 or newer.

[1]Dialog OS is a registered trademark of CLT Sprachtechnologie GmbH. Other product and company names listed are trademarks or trade names of their respective owners.

[2]Dialog OS is developed and distributed by CLT Sprachtechnologie GmbH: http://www.clt-st.de/dialogos

Default components Dialog OS comes with built-in modules for professional quality speech input and output using technology from Nuance and AT&T. As part of the platform, Dialog OS provides a number of default input/output device clients that can be directly connected without extra programming. Among those are: a simple text console for text-based input and output, a sound player, and a default client for a connection to an SQL database. CLT can also provide built-in connections to a number of other research and commercial Automatic Speech Recognition (ASR) and Text-To-Speech (TTS) systems.

Extensibility Dialog OS can be extended to work with an arbitrary number of clients through a Java-based API. The low-level communication between Dialog OS and the clients is handled by a dedicated internal protocol and remains invisible to the user. Programming a new client involves a Java implementation of a high-level functional protocol for the given client, without having to deal with the details of network connection with the dialogue engine itself.

FSA-based dialogue modelling The central part of the dialogue system is the dialogue model. Dialog OS offers an intuitive way of modelling dialogues using Finite State Automata (McTear, 2002). Building a dialogue model consists of adding and linking dialogue graph nodes represented as icons on a GUI workspace. Those include input/output nodes and internal nodes, for example, to execute scripts, set and test variables, enter a sub-graph (i.e. execute a sub-automaton).[3] The dialogue model is stored in an XML format.

Dialog OS builds on the functionality of its predecessor, DiaMant (Fliedner and Bobbert, 2003). Below, we list some of the features taken over, extended or enhanced in Dialog OS:

[3]The expressive power of the dialogue models is effectively that of push-down automata.

User input The input nodes for text-based or spoken interaction allow to specify a list of expected input values; outgoing edges are created automatically. User input may be matched directly against the list, or against a regular expression. For spoken input via default ASR components, both the recognised string and the recognition confidences can be accessed.

Built-in data types Global variables can be of simple types (e.g. String, Integer, etc.) as well as more complex data structures of key-value pairs.

Scripting language Dialog OS includes an interpreter of a JavaScript-like scripting language for simple data manipulation functions, e.g., to match input against a regular expression. These can be integrated through a *Script* node.

Sub-automata The *Procedure* node allows for flexible and modular dialogue modelling. Recurring parts of the dialogue can be saved as individual parameterisable sub-automata, direct counterparts of sub-routines in programming languages.

Wizard-of-Oz (WOz) mode Dialog OS can be run in WOz mode (Fraser and Gilbert, 1991) in which one or more of the "Devices" are simulated and dialogue execution details are saved in logfiles; this allows to set up small-scale WOz experiments.

3 Dialog OS in the classroom

We have been using Dialog OS and its predecessor at Saarbrücken in a number of courses involving spoken dialogue systems. Notable features that make it suitable for educational purposes include:

Intuitive interface: Learning to use Dialog OS takes very little time. Thanks to the GUI, even non-computational students can easily configure a functional system with little (or even no) knowledge of programming. The low learning overhead allows to concentrate on modelling interesting dialogue phenomena rather than technical details.

High-level language for building new components: A Java-based API makes the development process efficient and allows for the final system to be built on a single programming platform and kept highly modular.[4]

Below we briefly outline larger spoken dialogue systems developed as part of software projects using the Dialog OS framework.

Talking Robots with LEGO MindStorms® Within two runs of the course, students built various speech-enabled mobile robots using LEGO and Dialog OS as dialogue framework (Koller and Kruijff, 2004). Integration involved writing a client to control the MindStorms RCX (Dialog OS provides built-in support for MindStorms NXT). Luigi Legonelli, the Shell Game robot, and a modified version of Mico, the bar-keeper,[5] have been presented at CeBIT '03 and '06, respectively.

Campus information system A group of three students built a spoken information system for Saarland University campus. The system can answer questions on employee's offices, telephone numbers, office locations, etc. The highlights of the system are modularity[6] and an adaptive clarification model needed to handle many foreign names and foreign user accents.

Talking elevator In two editions of this course, students built speech interfaces to the elevators in the institute's buildings. In the first course, a simple mono-lingual system was developed. In an ongoing project, students are building a trilingual system with speaker identification, using their own version of a Nuance client and an elevator client that communicates with the elevator hardware via a serial protocol.

References

G. Fliedner and D. Bobbert. 2003. DiaMant: A Tool for Rapidly Developing Spoken Dialogue Systems. In *Proc. of DiaBruck*, pages 177–178, Wallerfangen, Germany.

N. M. Fraser and G. N. Gilbert. 1991. Simulating speech systems. *Computer Speech and Language*, 5:81–99.

A. Koller and G-J. M. Kruijff. 2004. Talking robots with lego mindstorms. In *Proc. of COLING-04*, pages 336–342.

O. Lemon and X. Liu. 2006. DUDE: A dialogue and understanding development environment, mapping business process models to information state update dialogue systems. In *Proc. of EACL-06*, pages 99–102, Trento, Italy.

M. F. McTear. 1999. Using the CSLU toolkit for practicals in spoken dialogue technology. In *Proc. of ESCA/SOCRATES Workshop on Method and Tool Innovations for Speech Science Education*, pages 113–116, UK.

M. F. McTear. 2002. Spoken dialogue technology: enabling the conversational user interface. *ACM Computing Surveys (CSUR)*, 34(1):90–169.

[4] A GUI is also part of CSLU (McTear, 1999) and DUDE (Lemon and Liu, 2006) dialogue toolkits. However, DUDE has not yet been tested with novice users, while extending CSLU Toolkit involves programming in C, rather than in a higher-level language such as Java.

[5] Mico's mechanics were substantially re-designed by CLT, however, the dialogue model was, for the most part, taken over from the student project.

[6] Currently, the system supports only German, but the modular design was motivated by anticipated extensions for English and French.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 161–162.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Complex Taxonomy Dialogue Act Recognition with a Bayesian Classifier

Mark Fishel
Dept of Computer Science
University of Tartu
Tartu, Estonia
fishel@ut.ee

1 Introduction

This paper describes the experiments of performing dialogue act (DA) recognition with a complex DA taxonomy using a modified Bayes classifier.

The main application of DA recognition is in building dialogue systems: classifying the utterance and determining the intention of the speaker can help in responding appropriately and planning the dialogue. However, in this work the target application is human communication research: with tagged DAs it is easier to search for utterances of a required type in a dialogue corpus, to describe the dialogues with a general model of dialogue moves, etc.

The DA taxonomy, used in the current work, was designed for the Estonian Dialogue Corpus (EDiC) (Hennoste et al., 2003). This means two additional difficulties for DA recognition. Firstly, DA taxonomies used for human communication research are as a rule much more detailed than in case of dialogue systems (e.g., comparing DCIEM (Wright, 1998) and CallHome Spanish (Ries et al., 2000) taxonomies); therefore, more DAs have to be distinguished, with several of them having unclear meaning boundaries. Secondly, Estonian is an agglutinative language with 14 cases, a complex system of grouping and splitting compound nouns, heterogeneous word order and several other features that make natural language processing harder.

2 Experiments

2.1 Experiment Setup

In order to determine the optimal set of input features additive feature selection was applied. All of the tests were performed using 10-fold cross-validation.

In this work we only tried simple features, not involving morphological analysis, part-of-speech tagging, etc. The used ones included DA tag bi- and trigrams, keywords and the total number of words in the utterance. Keyword features included the 1st word, first 2 words and first, middle and last words as a single dependency. We also tried stemming the words and alternatively leaving only the first 4 characters of the word.

The learning model used in this work is the Bayes classifier. Its original training/testing algorithm supports only a fixed number of input features. This makes it harder to include information with variable size, such as the set of the utterance words. In order to overcome this limitation, we slightly modified the algorithm by calculating the geometrical average of the conditional probabilities of the DA tag, given each utterance word. With this approach the probabilities remain comparable despite the variable length of the utterances.

The corpus used for training and testing is described in greater detail in (Gerassimenko et al., 2004), updated information can be found online[1]. The version used in the experiments contains 822 dialogues (a total of 32860 utterances) of mixed content (telephone conversations in an information service, at a travelling agency, shop conversations, etc).

2.2 Results

After the feature selection process converged, the following features were included into the selection:

[1] http://math.ut.ee/~koit/Dialoog/EDiC

DA tag trigram probabilities, the geometrical mean of the word-tag conditional probabilities and the number of words in the utterance. Stemming was not performed in the final preprocessing.

The resulting cross-validation precision over the whole set of dialogues was 62.8% with the resulting feature set. In general the most typical DA tag to be confused with was the most frequent one. In addition, some tags were frequently confused with each other.

In addition to the objective precision estimation provided by cross-validation, we also wanted to have a direct comparison of the resulting DA tagger with the human taggers. For that we applied the tagger to both human tagged parts, used for calculating the human agreement. The resulting precisions for the two parts are 80.5% and 78.6%.

3 Discussion

It is interesting to note that the resulting selection of representation features included only simple text-based features. Although the task of DA recognition belongs to computational pragmatics in natural language processing, in this case it gets solved on the level of pure text, which is even lower than the morphology level.

Future work includes several possibilities. In particular, several output errors of the trained classifier seem obvious to solve to a human tagger. For instance, several utterances containing wh-words are misclassified as something other than wh-questions. There are at least two possibilities to treat that kind of problems. Firstly, a set of rules can be composed by professional linguists to target each output problem individually. This approach has the advantage of guaranteed improvement in the required spot; on the other hand, manually composing the rules can result in overlooking some global influences on the remaining utterance cases, which can cause decreased performance in general. Another way to address the output errors would be to add more descriptive features to the input.

4 Conclusions

We have described a set of experiments, aimed at applying a Bayes classifier to dialogue act recognition. The targeted taxonomy is a complex one, including a large number of DA tags.

Additive feature selection was performed to find the optimal set of input features, representing each utterance. The tested features included n-gram probabilities and keyword-based features; the latter were tested both with and without stemming.

The resulting precision of the trained model, measured with 10-fold cross-validation is 62.8%, which is significantly higher than previously achieved ones. The selected features included DA tag trigram probabilities, number of words probability and the geometrical mean of the word-tag conditional probabilities of all the utterance words.

The model was compared to the agreement of human taggers in the targeted taxonomy – this was done by applying it to the same test corpus that was used in calculating the agreement. The two resulting precisions are 80.5% and 78.6%, which is very much near the human agreement (83.95%).

There is much room for further development of the classifier. This includes adding more specific features to the model's input, manually composed output post-processing rules, etc.

References

Olga Gerassimenko, Tiit Hennoste, Mare Koit, Andriela Rääbis, Krista Strandson, Maret Valdisoo, and Evely Vutt. 2004. Annotated dialogue corpus as a language resource: An experience of building the estonian dialogue corpus. In *Proceedings of the 1st Baltic Conference "Human Language Technologies. The Baltic Perspective"*, pages 150–155, Latvia, Riga.

Tiit Hennoste, Mare Koit, Andriela Rääbis, Krista Strandson, Maret Valdisoo, and Evely Vutt. 2003. Developing a typology of dialogue acts: Tagging estonian dialogue corpus. In *Proceedings of the 7th Workshop on the Semantics and Pragmatics of Dialogue, DiaBruck 2003*, pages 181–182, Saarbrücken, Germany.

Klaus Ries, Lori Levin, Liza Valle, Alon Lavie, and Alex Waibel. 2000. Shallow discourse genre annotation in callhome spanish. In *Proceecings of the International Conference on Language Ressources and Evaluation (LREC-2000)*, Athens, Greece.

H. Wright. 1998. Automatic utterance type detection using suprasegmental features. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'98)*, volume 4, page 1403, Sydney, Australia.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 163–164.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Default preferences in *donkey* anaphora resolution

Francesca Foppolo
Dept. of Psychology - University of Milano Bicocca
v.le dell'Innovazione 11
20126 Milano (Italy)
francesca.foppolo@unimib.it

Abstract

I will present an experimental study on the interpretation of pronouns in donkey sentences, i.e. sentences such as "Every farmer who owns a donkey beats *it*" that admits of two interpretations: the *universal* (= Every farmer who owns a donkey beats all the donkeys he owns) or the *existential* interpretation (=Every farmer who owns a donkey beats one of the donkeys he owns). By means of two reaction time experiments I show: (i) that the distribution of the two interpretations is the one predicted by Kanazawa's generalization (1994): the interpretation of donkey pronouns seems to be sensitive to the left monotonicity properties of the head determiner (Experiment 1); (ii) that such interpretations seem to be a matter of preference, i.e. a default that comes about in relatively "neutral" contexts and that appropriate context manipulations can override (Experiment 2).

1 Introduction

I will present an experimental study conducted with Italian adults concerning the interpretation of pronouns in donkey sentences. Consider the standard example in (1):

(1) Every farmer who owns a donkey beats *it*

As is well known from the literature, the pronoun *it* in (1) admits of two interpretations, the *universal* (∀) one and the *existential* (∃) one, interpretations whose truth conditional import can be represented as in (2) and (3) respectively :

(2) ∀-reading:
$\forall x\ [[\text{farmer}(x) \wedge \exists y\ \text{donkey}(y) \wedge \text{has}(x,y)] \rightarrow \forall z\ [\text{donkey}(z) \wedge \text{has}(x,z) \rightarrow \text{beats}(x,z)]]$
= *Every farmer who owns a donkey beats all the donkeys he owns*

(3) ∃-reading:
$\forall x\ [[\text{farmer}(x) \wedge \exists y\ \text{donkey}(y) \wedge \text{has}(x,y)] \rightarrow \exists z\ [\text{donkey}(z) \wedge \text{has}(x,z) \wedge \text{beats}(x,z)]]$
= *Every farmer who owns a donkey beats one of the donkeys he owns*

There are many proposals as to how these readings come about. However, our concern here is not so much to choose among such proposals (though eventually, we believe that our results will be relevant to such an issue). Our immediate concerns here are rather to experimentally test an interesting generalization regarding the distribution of ∀- and ∃-interpretations, put forth in Kanazawa (1994). According to Kanazawa, the preferred interpretation of donkey pronouns is the one that preserves the monotonicity properties of the determiner. This makes the following predictions on the sample set given in (4).

(4)

Det.	Monotonicity	interpretation
Every	↓↑	∀
No	↓↓	∃
Some	↑↑	∃

Kanazawa's point, to whose work we must refer for details, is that the interpretations in the last column in (4) are the only ones that preserve (in a donkey anaphora context) the monotonicity properties of each lexical determiner, spelled out in the second column. While there has been some experimental work on how donkey pronouns are interpreted (cf., e.g. Geurts, 2002), no work has tried to experimentally probe Kanazawa's claim. Yet, if empirically supported, such a claim would be important, as it would show that the semantic processor must have access to an abstract formal property of an unprecedented kind (namely, monotonicity preservation in non C-command anaphora).

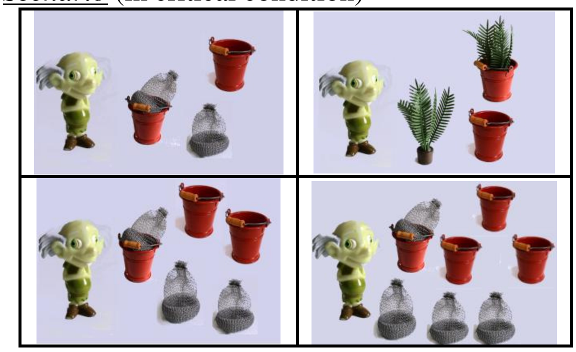
2 The experimental study

1.2 Material and Procedure

We carried out a reaction-time study with a total of 66 Italian-speaking adults. Subjects were asked to evaluate *donkey sentences* introduced by different types of quantifiers with respect to scenarios displaying four pictures. Sentences were presented in two critical conditions: in the absence of an extra-linguistic context (Exp.1) and after the addition of a biasing context (Exp. 2). In both cases, to avoid interferences from extra-linguistic knowledge, we used strange characters (introduced as aliens) with weird objects to which only fantasy names were given. A sample of critical sentences used is given in (5)-(7), and one of the scenarios proposed is presented next:

(5) *Every Flont that has a vilp keeps it in a bin*
(6) *No Flont that has a vilp keeps it in a bin*
(7) *Some Flont that has a vilp keeps it in a bin*

Scenario (in critical condition)



Note that, given that two alternative interpretations can be associated to each sentence (as shown is (2) and (3) above), the scenario above makes the critical sentences true under one interpretation, but crucially false under the other. In case of Exp. 2, a biasing context was added before the same scenario appeared, in the aim of inducing subjects to accept the *donkey sentence* under the reading predicted as dispreferred by Kanazawa's generalization.

2.2 Results

Subjects' answers in Exp. 1 seems to conform to the predictions derived from the generalization in (4), at least in case of *Some* and *No*: in both cases, the reading that emerged as preferred in the critical condition was the *existential* one (87% and 93% in case of *Some* and *No* respectively). In case of *Every*, instead, subjects split. However, this result is compatible with the results obtained in Exp. 2, which show that subjects do in fact access the alternative interpretation of the anaphora, but crucially that its availability varies in accordance with the initial head determiner: the dispreferred (∃) reading is very easily accessed in case of *Every* (a significantly higher proportion of subjects (i.e. 81%) judged sentence (5) TRUE in the scenario above in Exp. 2). Conversely, the access to the dispreferred (∀) interpretation of the anaphora is much harder in case of sentences (6) and (7), even in presence of a context that biases subjects towards this interpretation.

3 Conclusion

Two main points emerge from our results. First, Kanazawa's generalization does appear to be empirically supported. How donkey pronouns are interpreted seems to be sensitive to the monotonicity properties of the determiners involved along the lines indicated in (4). Second, such interpretations seem to be a matter of preference (i.e. a default that comes about in relatively "neutral" contexts). As Exp. 2 shows, appropriate context manipulations lead to the emergence of the alternative interpretation. These results illustrate several general points. For one thing, they show that speakers unconsciously and systematically compute abstract properties pertaining to entailment patterns, as they tend to choose the interpretation of the donkey pronouns that retains the lexical properties of the determiner. Work on negative polarity has arguably shown sensitivity to monotonicity patterns in determining the distribution of items like *any*. Here we detect a similar phenomenon in connection with a purely interpretive task (namely, how pronoun readings in non C-command anaphora are accessed). This paves the way for further research (e.g., with respect to figuring out *how* various readings come about, and with respect to testing the present claim with other determines and settings) and confirms the value of integrating theoretical claims in semantics with experimental work.

Selected References.

Chierchia, G. (1995). *Dynamics of meaning: anaphora, presupposition, and the theory of grammar*. Chicago, University of Chicago Press.

Kanazawa, M. (1994). Weak vs. Strong Readings of Donkey Sentences and Monotonicity Inference in a Dynamic Setting. *Linguistics and Philosophy* 17: 109-158.