SLTC 2010

The Third Swedish Language Technology Conference (SLTC 2010)

Linköping

October 27-29, 2010

Proceedings of the Conference

Organizing committee

Lars Ahrenberg, Lise-Lott Andersson, Maria Holmqvist, Arne Jönsson, Magnus Merkel, Sara Stymne, Linköping University;

Sture Hägglund, Santa Anna IT Research

Mats Wirén, Stockholm University,

Rickard Domeij, Language Council of Sweden.

The organizers gratefully acknowledge support from the Swedish Graduate School in Language Technology (GSLT)

Program

Wednesday 27. October, 13.15 – 17.00, in **Visionen**

Nordic seminar on language technology and accessibility

Webbtillgänglighet i flerspråkigt perspektiv,
Rickard Domeij, Språkrådet i Sverige.

Tilgjengelig samisk og andre nasjonale minoritetsspråk,
Sjur Nørstebø Moshagen, Sametinget i Norge.

Døves behov for og brug af informationer på Internettet,
Lene Schmidt, Center för døvblindhed og høretab, Danmark.

Coffee Break (Ljusgården)

Teknologi og lydbaserte brukergrensesnitt – behovet for talegjenkjenning, lydstyring og syntetiske barnestemmer,
Morten Tollefsen, MediaLT, Norge.

Tillgänglighet till terminologi – svenska myndigheters ansvar,
Henrik Nilsson och Magnus Merkel, Terminologicentrum TNC i Sverige resp. Linköpings universitet.

Automatisk översättning på svenska myndigheters webbplatser,
Stefan Johansson, Funka Nu, Sverige.

Molto - automatisk översättning att lita på?
Aarne Ranta, Göteborgs universitet.

Thursday 28. October, 09.00 – 12.00 in **Visionen**

Nordic seminar on language technology and accessibility

Forbedret tilgjengelig informasjon og service på nettet med PipeOnline og Brage Olav Indergaard, NLB. (Följs av 15 min samtal med företrädare för svenska TPB och danska Nota).

Ökad tillgänglighet för teckenspråkiga med teckenspråksteknologi,
Björn Granström och Johan Beskow, KTH.

Coffee Break (Ljusgården)

Læse-og skrivestøtte for bedre tilgængelighed,
Søren Aksel Sørensen, Mikroverkstedet.

Paneldiskussion och avslutning

Thursday 28. October, 12.00 – 13.00 in **Ljusgården**
Sandwich lunch

Thursday 28. October, 13.00 – 14.55 in **Visionen**
General session 1

Opening

Invited speaker: Jan Alexandersson, DFKI, Saarbrücken

”The DFKI Competence Center for Ambient Assisted Living: Some Activities and Visions.”

Coffee Break (Ljusgården)

Thursday 28. October, 15.15 – 16.15 in **Visionen**
General session 2

Thursday 28. October, 15.15 – ca 17.15 in **John von Neumann**
Workshop 1: Readability and multilingualism

Thursday 28. October, 16.15 – ca 18.00 in and around **Visionen**
Poster session

The poster session starts with a round of poster puffs in Visionen.

Thursday 28. October, 18.30 - late in **Ljusgården**
Conference dinner

Friday 29. October, 09.00 – 10.00 in **Visionen**
General session 3

Invited speaker: Keith B. Hall, Google Research, Zürich

”Language technology at Google”

Coffee Break (Ljusgården)

Friday 29. October, 10.20 – 12.00 in **Visionen**
General session 4

Short break

Announcements and break

Friday 29. October, 13.15 – 15.15 in **John von Neumann**
Workshop 2: Compounds and multiword expressions

Friday 29. October, 13.15 – 16.30 in **Grace Hopper**
Workshop 3: Svensk dialogverkstad

Pay no attention to … eh … that man behind the curtain

Gabriel Skantze, Anna Hjalmarsson
Department of Speech, Music and Hearing
KTH, Stockholm
gabriel@speech.kth.se, annah@speech.kth.se

Abstract

We present an experimental study that explores an early implementation of a model of speech generation for incremental dialogue systems. The model allows a dialogue system to incrementally interpret spoken input, while simultaneously planning, realising and self-monitoring the system response. The model has been implemented in a general dialogue system framework. Using this framework, we have implemented a specific application and tested it in a Wizard-of-Oz setting, comparing it with a non-incremental version of the same system. The results show that the incremental version, while producing longer utterances, has a shorter response time and is perceived as more efficient by the users.

1. Introduction

Speakers in dialogue understand and produce speech incrementally as the dialogue progresses, using information from several different sources to decide what to say and when it is appropriate to speak (Levelt, 1989). While speaking, processes at all levels – semantic, syntactic, phonologic and articulatory – work in parallel to render the message under construction. This is an efficient processing strategy since speakers may employ the time devoted to articulating the first part of a message to plan the rest. Contrary to this, most spoken dialogue systems use a silence threshold to determine when the user has stopped speaking. The user utterance is then processed by one module at a time, after which a complete system utterance is produced and realised by a speech synthesizer.

This paper presents a study that explores how incremental speech generation can be used in a Wizard-of-oz setting to improve the response time. A model of incremental speech generation has been implemented that allows the dialogue system to incrementally interpret spoken input, while simultaneously planning, realising and self-monitoring the system response.

2. Incremental processing

The proposed model is based on a general, abstract model of incremental processing proposed by Schlangen & Skantze (2009) and has been implemented in Jindigo – a Java-based open source framework for implementing and experimenting with incremental dialogue systems, developed at KTH (www.jindigo.net). We only have room for a very brief overview of the model here, but interested readers are referred to Skantze & Hjalmarsson (in press). We currently use a typical pipeline architecture for the dialogue system (see Figure 2, in which a Wizard is used instead of an ASR). Contrary to most dialogue systems, input and output is not processed and produced utterance by utterance, but instead on the level of words and sub-phrases. An example is shown in Figure 1. As the words are incrementally recognized by the ASR, they are processed by each dialogue system component, and a SpeechPlan with possible responses (represented as a graph) is incrementally produced by the ActionManager. If the system detects that the user has finished speaking and it is appropriate for the system to start speaking, the Vocalizer may start realising the SpeechPlan, even if it is not yet complete. The ActionManager may also revise the SpeechPlan if needed, for example if a speech recognition hypothesis turns out to be incorrect in light of more speech input. The Vocalizer can then automatically make covert or overt self-repairs, i.e. either without the user noticing it, or using an editing term, such as "sorry, I mean". If it is appropriate for the system to start speaking and the SpeechPlan has not yet been constructed, the Vocalizer may also use filled pauses such as "eh".

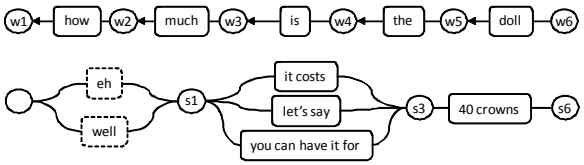


Figure 1: The output of an ASR (top) and the SpeechPlan that is incrementally produced (bottom). Vertex s1 is associated with w1, s3 with w3, etc.

3. A Wizard-of-Oz experiment

A Wizard-of-Oz experiment was conducted to test the usefulness of the model outlined above. All modules in the system were fully functional, except for the ASR, since not enough data had been collected to build language models. Thus, instead of using ASR, the users' speech was transcribed by a Wizard. A common problem is the time it takes for the Wizard to transcribe incoming utterances, and thus for the system to respond. With the proposed model for incremental speech generation, the system may start to respond even if the Wizard has not yet completed the transcription.

The experimental setup is shown in Figure 2. A standard Voice Activity Detector (VAD) is used to detect the end of the user's utterance and trigger the Vocalizer to

start speaking. The Wizard may start to type as soon as the user starts to speak and may alter whatever he has typed until the return key is pressed and the hypothesis is committed.

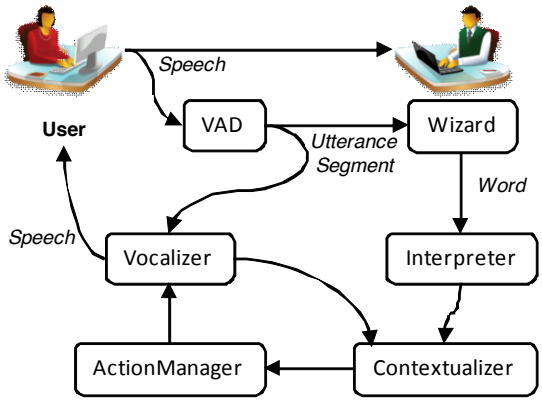


Figure 2: The system architecture used in the Wizard-of-Oz experiment.

We used a spoken dialogue system for second language learners of Swedish under development at KTH, called DEAL (Wik & Hjalmarsson, 2009). The scene of DEAL is set at a flea market where a talking agent is the owner of a shop selling used goods. The shop-keeper can talk about the properties of goods for sale and negotiate about the price. For the experiment, DEAL was re-implemented using the Jindigo framework.

An experiment with 10 participants, 4 male and 6 female, was conducted to compare the incremental implementation of DEAL to a non-incremental version of the same system. The participants were given a mission: to buy three items (with certain characteristics) in DEAL at the best possible price from the shop-keeper. The participants were further instructed to evaluate two different versions of the system, System A and System B. However, they were not informed how the versions differed. The participants were lead to believe that they were interacting with a fully working dialogue system and were not aware of the Wizard-of-Oz set up. Post experiment questionnaires were used to explore which one of the two versions was most prominent according to 8 different dimensions: which version they *preferred*; which was more *human-like*, *polite*, *efficient*, and *intelligent*; which gave a *faster response* and better *feedback*; and with which version it was easier to know *when to speak*.

4. Results

A video (with subtitles) showing an interaction with one of the users can be seen at http://www.youtube.com/watch?v=cQQmgItIMvs. Figure 3 shows the difference in response time between the two versions. As expected, the incremental version started to speak more quickly (*M*=0.58s, SD=1.20) than the non-incremental version (*M*=2.84s, SD=1.17), while producing longer utterances.

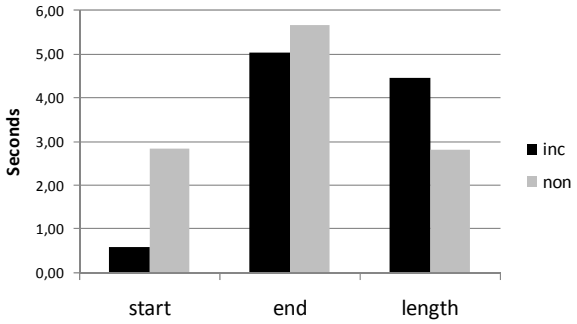


Figure 3: The first two column pairs show the average time from the end of the user's utterance to the start of the system's response, and from the end of the user's utterance to the end of the system's response. The third column pair shows the average total system utterance length (end minus start).

It was harder to anticipate whether it would take more or less time for the incremental version to finish utterances. The incremental version initiates utterances with speech segments that contain little semantic information. Thus, if the system is in the middle of such a segment when receiving the complete input from the Wizard, the system may need to complete this segment before producing the rest of the utterance. On the other hand, it may also start to produce speech segments that are semantically relevant, based on the incremental input, which allows it to finish the utterance more quickly. As the figure shows, it turns out that the average response completion time for the incremental version (*M*=5.02s, SD=1.54) is about 600ms faster than the average for non-incremental version (*M*=5.66s, SD=1.50), ($t(704)=5.56$, $p<0.001$).

To analyze the results of the questionnaire, a Wilcoxon Signed Ranks Test was carried out. The results show that the two versions differed significantly in three dimensions, all in favour of the incremental version. Hence, the incremental version was rated as more *polite*, more *efficient*, and *better at indicating when to speak*.

The experiment shows that it is possible to achieve fast turn-taking and convincing responses in a Wizard-of-Oz setting. We think that this opens up new possibilities for the Wizard-of-Oz paradigm, and thereby for practical development of dialogue systems in general.

5. References

Levelt, W. J. M. (1989). *Speaking: From Intention to Articulation*. Cambridge, Mass., USA: MIT Press.

Schlangen, D., & Skantze, G. (2009). A general, abstract model of incremental dialogue processing. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL-09)*. Athens, Greece.

Skantze, G., & Hjalmarsson, A. (in press). Towards Incremental Speech Generation in Dialogue Systems. To be published in *Proceedings of SigDial*. Tokyo, Japan.

Wik, P., & Hjalmarsson, A. (2009). Embodied conversational agents in computer assisted language learning. *Speech communication, 51*(10), 1024-1037.

A Maximum Latency Classifier for Listener Responses

Daniel Neiberg[1], **Khiet P. Truong**[2]

[1]Department of Speech, Music and Hearing, Royal Instituite of Technology (KTH), Sweden
[2] Human Media Interaction, University of Twente, The Netherlands
[1]neiberg@speech.kth.se, [2]k.p.truong@ewi.utwente.nl

Abstract

When Listener Responses such as "yeah", "right" or "mhm" are uttered in a face-to-face conversation, it is not uncommon for the interlocutor to continue to speak in overlap, i.e. before the Listener becomes silent. We propose a classifier which can classify incoming speech as a Listener Response or not before the talk-spurt ends. The classifier is implemented as an upgrade of the Embodied Conversational Agent developed in the SEMAINE project during the eNTERFACE 2010 workshop.

1. Introduction

Face-to-face conversation in the map-task domain may be viewed as a role play where one is having the Speaker role while the other is having the Listener role. Being an attentive speaker involves creating opportunities for the listener to give listener responses, such as "yeah", "right", "mhm" or head-nodes and other gestures and continue speaking at the appropriate moment.

The Listener commonly utter responses such as "yeah", "mhm", "uhu". Fujimoto (Fujimoto, 2007) propose to call these short utterances Listener Responses. These are short utterances or vocalizations which are interjected into the Speakers' account without causing an interruption, or being perceived as competitive of the floor.

In this work, we show the presence of a negative gap (overlap) between the continuation talk-spurt of the Speaker following the onset of a Listener Response. Based on this insight, we propose a detector which is able to classify incoming speech as Listener Responses before the talk-spurt ends.

2. The MapTask Corpus

The HCRC Map Task Corpus (Anderson et al., 1991) contains 128 dialogues. The task is for one subject to explain a route to another subject. We use the half of the dialogs which were recorded under a face-to-face condition. The two conversations labeled as `q3ec1` and `q3ec5` were discarded due to a buzz in the speech signal.

We used the official MapTask annotations concerning the distinction between Acknowledgment Moves (MTACK) and other talkspurts (NONMTACK). The precise definition of an Acknowledgment Move is found in (Carletta et al., 1997), which closely resemble the term Listener Response and thus serve our purpose. The inter-label agreement of the Map Task Corpus annotations are good ($\kappa = .83$).

Based on the provided annotations, the corpus is segmented into *talkspurts* (Brady, 1968), defined as a minimum voice activity duration of 50ms separated by a minimum inter-pause of 200 ms. The inter-pause threshold correspond to the minimally perceived silence, and the resulting segmentation will better resemble the condition when a real voice activity detector is used.

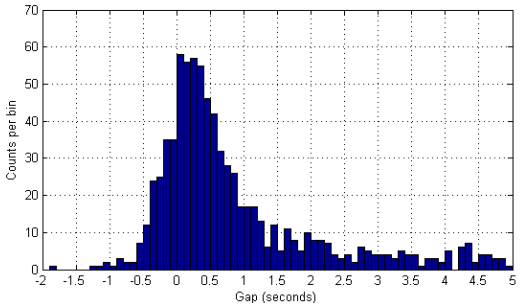


Figure 1: The gap or overlap (negative gap) between a MTACK Response and the incterlocutors' continuation using bins of 100 ms.

2.1 The gap or overlap after Listener Responses

Figure 1 shows the distribution of gaps between talkspurts of the Speaker which follows the onset of a MTACK Response. This gap has a negative value (i.e. overlap) if the Speaker continues speaking before the end of the Response. Although the graph shows the Speaker commonly continues to speak after roughly 0-400 ms, it also shows that overlap is not uncommon. This means that for a responsive dialog with a Virtual Human, Responses from the user need to be classified before they are finished. This might be done using a speech recognizer running in incremental mode or by using a specialized detector. Since a speech recognizer will only detect lexical content, the special prosodic characteristics of listener responses cannot be accounted for.

3. Maximum latency classification

Based on the analysis in the previous Section, we propose a maximum latency design for the detector. It is implemented as a voice activity detector which sends an end message after the talkspurt ends, or at a predefined duration threshold, denoted as the maximum latency. If the duration reaches the threshold, it continues to work as normal voice activity detector internally, otherwise it might trigger again. Note that the detector may trigger before the maximum latency if the talkspurt is shorter than the threshold subtracted by the minimum inter pause threshold.

3.1 Feature trajectories as length-invariant Discrete Cosine Coefficients

To parameterize the trajectories of each feature through out a talkspurt, we use DCT coefficients invariant to segment length:

$$X_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}(n+\frac{1}{2})k\right) \qquad k = 0,\ldots,N$$

where N is the segment length, x_n is the feature value at time n and X_k is the k'th coefficient.

These DCT coefficients are much faster to compute than polynomial regression coefficients, since polynomial regression require matrix inversion. The 0'th coefficient is equal to the arithmetic average, which means if it is omitted, then only the relative shape of a trajectory is parametrized. This property is useful for parameterizing features such as F0 (which has a speaker dependent additive bias) or MFCCs (which has an additive channel bias). When a DCT is applied on MFCCs, one obtain the cepstrum modulation spectrum. The usage of length-invariant cepstrum modulation spectrum was first introduced by (Ariki et al., 1989), although no specific term was used at the time.

Back-channels has been shown to have a rise in F0 as well as have distinct intensity contours (Benus et al., 2007). Other important Listener Response characteristics are lexical content and short duration (Edlund et al., 2010). This makes us come up with the feature set: F0 ENVELOPES, INTENSITY, MFCCS, DURATION, (For training, the full talkspurt duration was used, for testing, the duration up to the maximum latency threshold was used) and SPECTRAL FLUX (the L2-norm of FTT-bins in adjacent frames). All feature are parametrized in the time dimension using length invariant DCT-coefficients 1-6, except SPECTRAL FLUX for which we use 0-5, unless anything else is specified.

4. Experiments

For all experiments, the training set consists of so-called quads 1-4, the development set holds quads 5-6 and the evaluation set holds quads 7-8. This gives us around 500-1000 MTACK and NONMTACK talkspurts per set. For classification, we used Support Vector Machines with Radial Base kernel as implemented in the LibSVM package (Chang and Lin, 2001). The SVM regularization parameters ν and γ are optimized on the development set, and the model with the best parameters is then used on the evaluation set.

5. Results And Discussion

As expected, we observe in Table 1 that MFCCs and duration, at least in the 500 ms case, are the main contributors to the distinction between MTACK vs. NONMTACK, while F0 is the weakest feature. We observe that omitting the 0th DCT for MFCCs, does not hurt performance. For the 300 ms latency, this leads to a feature combination of Int., Sp. flux, MFCC without 0th, while for the 500 ms latency, we add duration. These two classifiers are then tested on the evaluation set, as shown in the Table.

Development set		
Feature(s)	300 ms	500 ms
F0 Envelopes	55	59
Intensity	60	62
MFCC with 0th	72	75
MFCC without 0th	74	75
Duration	55	71
Spectral flux	66	67
Int., Sp. flux, MFCC with 0th	73	76
Int., Sp. flux, MFCC with 0th, Dur.	75	76
Int., Sp. flux, MFCC without 0th	74	76
Int., Sp. flux, MFCC without 0th, Dur.	73	76
Evaluation set		
Feature(s)	300 ms	500 ms
Int., Sp. flux, MFCC without 0th	73	
Int., Sp. flux, MFCC without 0th, Dur.		76

Table 1: Average F-scores in percent for "MTACK vs other" classification.

6. Conclusions and Acknowledgments

The good performance of the classifier at a maximum latency of 300 ms and 500 ms, made us decide to implement on-line versions of both. The research leading to these results has partly received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n 211486 (SEMAINE).

7. References

Anne H. Anderson, M Bader, E. G. Bard, E Boyle, Gwyneth Doherty-Sneddon, S. Garrod, Stephen Isard, Jacqueline C. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. Thompson, and R. Weinert. 1991. The HCRC Map Task corpus. *Language and Speech*, 34:351–366.

Y Ariki, S. Mizuta, M. Nagata, and T. Sakai. 1989. Spoken-word recognition using dynamic features analysed by two-dimensional cepstrum. *Communications, Speech and Vision*, 136(2):133–140, April.

S. Benus, Agustín Gravano, and J. Hirschberg. 2007. The prosody of backchannels in american english. In *Proceedings of the 16th International Congress of Phonetic Sciences 2007*, pages 1065–1068.

P T Brady. 1968. A statistical analysis of on-off patterns in 16 conversations. *The Bell System Technical Journal*, 47:73–91.

Jean C. Carletta, Stephen Isard, Gwyneth Doherty-Sneddon, Amy Isard, Jacqueline C. Kowtko, and Anne H. Anderson. 1997. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23(1):13–31.

C. C. Chang and C. J. Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

J. Edlund, M. Heldner, S. Al Moubayed, Agustín Gravano, and J. Hirschberg. 2010. Very short utterances in conversation. In *Proceedings of Fonetik*.

Donna T. Fujimoto. 2007. Listener responses in interaction: A case for abandoning the term, backchannel. *Journal of Osaka Jogakuin 2year College*, 37:35–54.

Swedish FrameNet++

**Lars Borin, Dana Dannélls, Markus Forsberg,
Maria Toporowska Gronostaj, Dimitrios Kokkinakis**

Språkbanken, Dept. of Swedish Language,
University of Gothenburg, Sweden
first.last@gu.se

Access to multi-layered lexical, grammatical and semantic information representing text content is a prerequisite for efficient automatic understanding and generation of natural language. A framenet built along the lines of the original English Berkeley FrameNet (see <http://framenet.icsi.berkeley.edu/>) is considered a valuable resource for both linguistics and language technology research that may contribute to the achievement of these goals.

Currently, framenet-like resources exist for a few languages, including some domain-specific and multilingual initiatives (Dolbey et al., 2006; Boas, 2009; Uematsu et al., 2009), but are unavailable for most languages, including Swedish, although there have been some pilot studies exploring the semi-automatic acquisition of Swedish frames (Johansson and Nugues, 2006; Borin et al., 2007).

At the University of Gothenburg, we have recently embarked on a project to build a Swedish framenet-like resource. A novel feature of this project is that the Swedish framenet will be an integral part of a larger lexical resource containing much other lexical information in addition to the framenet part, including information relating to older stages of Swedish. Hence the name *Swedish FrameNet++* (SweFN++).

As a result of almost half a century of work on Swedish linguistic resources and Swedish lexicography, our research unit is the owner of a number of digital linguistic resources of various kinds – including both data and processing resources – with various degrees of coverage, and in various formats. When now starting the construction of a Swedish framenet, recycling as much as possible of the content of these hard-won resources will be a priority.

In addition, there are freely available suitable resources created elsewhere that can also be thrown into the pot. Below we describe briefly some of the existing lexical resources.

Resources at Gothenburg

Resources for modern Swedish

SALDO is the core lexicon of the SweFN++ to which all other information is to be merged. It provides morphological and lexical-semantic information on about 88,500 entries (senses expressed by single words or multi-word units). The lexicon is an updated version of *The Swedish Associative Thesaurus* (Lönngren, 1989) remade into a fully digital resource and enhanced by Borin and Forsberg (2009a).

The SIMPLE and PAROLE lexicons for Swedish are lexical resources aimed at language technology applications, results of the EU projects PAROLE (1996–1998) and SIMPLE (1998–2000) (Lenci et al., 2000). SIMPLE contains 8,500 semantic units being characterised with respect to semantic type, domain and selectional restrictions. All the items are also linked to the PAROLE lexicon, which contains 29,000 syntactic units representing syntactic valence information.

The Gothenburg Lexical Database (GLDB) is a lexical database for modern Swedish covering 61,000 entries with an extensive description of their inflection, morphology and semantics. SDB (Semantic Database) is a version of GLDB where many of the verb senses have been provided with semantic valence information using a set of about 40 general semantic roles (Järborg, 2001) and linked to example sentences in a corpus. One goal of the work presented here will be to find effective ways of correlating framenet frame elements with these general semantic roles.

Historical resources

Dalin's dictionary (appr. 63,000 entries) reflects the Swedish language of the 19th century (Dalin, 1853 1855). It has been digitized and published with a web search interface at Språkbanken.

It is currently being linked on the sense level to SALDO as part of an eScience collaboration with historians interested in using 19th century fiction as historical source material. A morphological analysis module for this historical language variety is also being developed as part of this effort.

Old Swedish dictionaries There are three major dictionaries of Old Swedish (1225–1526): (Söderwall, 1884) (23,000 entries), Söderwall supplement (Söderwall, 1953) (21,000 entries), and (Schlyter, 1887) (10,000 entries). All have been digitized by Språkbanken.

We have started the work on creating a morphological component for Old Swedish (Borin and Forsberg, 2009b), covering the regular paradigms and created a smaller lexicon with a couple of thousand entries.

Resources from outside sources

The People's Synonym Dictionary is the result of a collaborative effort where users of a Swedish-English online dictionary have been asked to judge the degree

of synonymity of a word pair (randomly chosen from a large set of synonym candidates) on a scale from 0 (no synonymy) to 5 (complete synonyms). The downloadable version contains all word pairs with a rating in the interval 3 to 5, almost 40,000 Swedish synonym pairs. A Swedish-English dictionary – *Folkets lexikon* 'the People's Dictionary' – is now being constructed by the same method.

Swedish Wiktionary at present contains almost 60,500 entries (subdivided into senses). Notably, for each sense there is a free-text definition provided. Definitions are rare in other free lexical resources, which makes Swedish Wiktionary interesting for our purposes.

The Lund University frame list Johansson and Nugues (2006) have performed several experiments in attempt to create a Swedish framenet automatically. One of their experiments has resulted in list of 17,844 Swedish lemmas annotated with the English frames they evoke. The data was produced through parallel corpora with classification accuracy of 75%.

Merging lexical resources

The available lexical resources are heterogeneous as to their content and coding. The resources have been developed for different purposes by different groups with different backgrounds and assumptions, some by linguists, some by language technology researchers – possibly with little linguistic background or none at all – and yet others in Wikipedia-like collective efforts. Thus one of the main challenges for SweFN++ is to ensure content interoperability not only among the lexical resources but also between the available tools for text processing and lexical resources to be used by various pieces of software, and to formulate strategies for dealing with the uneven distribution of some types of information in the resource (e.g., syntactic valence information at present being available for about one fourth of the entries). This is work that we have initiated quite independently of the SweFN++ plans, within the European infrastructure initiative CLARIN (See <http://www.clarin.eu>).

We envision the end product of this work as a diachronic lexical resource for Swedish, to be used in developing language technology tools for dealing with text material from all periods of the written Swedish language, i.e., from the Middle Ages onwards. It remains to be seen how much this can apply the framenet part of the resource, but realistically, in addition to the modern language, at least 19th century Swedish may be covered.

The current state of the project can be viewed at the project homepage: <http://spraakbanken.gu.se/swefn>. The content of the page is automatically updated daily, hence reflecting the project as-is. At the time of writing, the Swedish framenet contained 113 frames with 5,961 lexical units.

References

Hans C. Boas, editor. 2009. *Multilingual Framenets in Computational Lexicography*. Mouton de Gruyter, Berlin.

Lars Borin and Markus Forsberg. 2009a. All in the family: A comparison of SALDO and WordNet. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, NEALT Proceedings Series, Vol. 4 (2009), Odense, Denmark. Kristiina Jokinen and Eckhard Bick.

Lars Borin and Markus Forsberg. 2009b. Something old, something new: A computational morphological description of Old Swedish. In *LREC 2008 Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, pages 9–16, Marrakech. ELRA.

Lars Borin, Maria Toporowska Gronostaj, and Dimitrios Kokkinakis. 2007. Medical frames as target and tool. In *FRAME 2007: Building Frame Semantics resources for Scandinavian and Baltic languages.*, pages 11–18, University of Tartu. Nodalida.

Anders Fredrik Dalin. 1853–1855. *Ordbok öfver svenska språket. Vol. I—II.* Stockholm.

Andrew Dolbey, Michael Ellsworth, and Jan Scheffczyk. 2006. BioFrameNet: A domain-specific FrameNet extension with links to biomedical ontologies. In *CEUR Workshop Proceedings*.

Jerker Järborg. 2001. Roller i Semantisk databas. Technical Report GU-ISS-01-3, Department of Swedish Language, University of Gothenburg.

Richard Johansson and Pierre Nugues. 2006. A FrameNet-based semantic role labeler for Swedish. In *Proceedings of Coling/ACL 2006*, Sydney. ACL.

Alessandro Lenci, Nuria Bel, Federica Busa, Nicoletta Calzolari, Elisabetta Gola, Monica Monachini, Antoine Ogonowski, Ivonne Peters, Wim Peters, Nilda Ruimy, Marta Villegas, and Antonio Zampolli. 2000. SIMPLE: A general framework for the development of multilingual lexicons. *Lexicography*, 13(4):249–263, December.

Lennart Lönngren. 1989. Svenskt associationslexikon: Rapport från ett projekt inom datorstödd lexikografi. Rapport ucdl-r-89-1, Centrum för datorlingvistik, Uppsala universitet.

C.J. Schlyter. 1887. *Ordbok till Samlingen af Sweriges Gamla Lagar. (Saml. af Sweriges Gamla Lagar 13)*. Lund, Sweden.

Knut Fredrik Söderwall. 1884. *Ordbok Öfver svenska medeltids-språket. Vol I–III.* Lund, Sweden.

Knut Fredrik Söderwall. 1953. *Ordbok Öfver svenska medeltids-språket. Supplement. Vol IV–V*. Lund, Sweden.

Sumire Uematsu, Jin D. Kim, and Jun'ichi Tsujii. 2009. Bridging the gap between domain-oriented and linguistically-oriented semantics. In *Proceedings of the BioNLP 2009 Workshop*, pages 162–170, Boulder, Colorado, USA. ACL.

Frames in Formal Semantics

Robin Cooper

Department of Philosophy, Linguistics and Theory of Science, University of Gothenburg
Box 200, S-405 30 Göteborg
cooper@ling.gu.se

1. Introduction

In his classic paper on frame semantics, Fillmore (1982) says:

> Frame semantics comes out of traditions of empirical semantics rather than formal semantics. It is most akin to ethnographic semantics, the work of the anthropologist who moves into an alien culture and asks such questions as, 'What categories of experience are encoded by the members of this speech community through the linguistic choices that they make when they talk?'

In this paper (a version of which has appeared as Cooper (2010)), we will make a connection between formal semantics and frame semantics by importing into our semantic analysis objects which are related to the frames of FrameNet. Our way of doing this will be different from, for example, Bos and Nissim (2008). An important part of our proposal will be that we introduce semantic objects corresponding to frames and that these objects can serve as the arguments to predicates. We will use record types as defined in TTR, type theory with records, (Cooper, 2005a; Cooper, 2005b; Cooper, forthcoming; Ginzburg, forthcoming) to characterize our frames. The advantage of records is that they are objects with a structure like attribute value matrices as used in linguistics. Labels (corresponding to attributes) in records allow us to access and keep track of parameters defined within semantic objects. This is in marked contrast to classical model theoretic semantics where semantic objects are either atoms or unstructured sets and functions. We will first show how TTR can be used to represent frames. We will then show how we propose to represent the contents of verbs in a compositional semantics. The use of frames here leads us naturally from the Priorean tense operators used by Montague to the Reichenbachian account of tense (Reichenbach, 1947) preferred by most linguists. The use of frames also leads us to a particular view of Partee's puzzle about temperature and price first discussed in Montague (1973) (PTQ, reprinted as Chap. 8 of Montague (1974)). Our solution to this puzzle relates to Fernando's (Fernando, 2006; Fernando, 2009) theory of events as strings of frames. Finally, we will consider how our proposal can be used to talk about how agents can modify word meaning by adjusting the parameters of word contents. This relates to a view of word meaning as being in a constant state of flux as we adapt words to describe new situations and concepts.

2. Using TTR to represent frames

Consider the frame `Ambient_temperature` defined in the Berkeley FrameNet. Leaving out a number of frame elements, we will say that an ambient temperature frame is a record of type (1).

(1)
$$\begin{bmatrix} \text{x} & : & \textit{Ind} \\ \text{e-time} & : & \textit{Time} \\ \text{e-location} & : & \textit{Loc} \\ \text{c}_{\text{temp\_at\_in}} & : & \text{temp\_at\_in(e-time, e-location, x)} \end{bmatrix}$$

We will call this type *AmbTemp*.

3. A TTR approach to verbs in compositional semantics

If you look up *run* on FrameNet you will find that on one of its readings it is associated with the frame `Self_motion`. Like many other frames in FrameNet this has a frame element `Time` which in this frame is explained as "The time when the motion occurs". This is what Reichenbach (Reichenbach, 1947) called more generally *event time* and we will use the label 'e-time'.

In order to obtain the content of the verb *ran* we need to create a function which abstracts over the individual which is to be its subject argument. Because frames will play an important role as arguments to predicates below we will not abstract over individuals but rather over frames containing individuals. The content of the verb *ran* will be (2).

(2)
$$\lambda r{:}\left[\text{x}{:}\textit{Ind}\right]\left(\begin{bmatrix} \text{e-time} & : & \textit{TimeInt} \\ \text{c}_{\text{tns}} & : & \text{e-time.end} < \iota\text{.start} \\ \text{c}_{\text{run}} & : & \text{run}(r\text{.x,e-time}) \end{bmatrix}\right)$$

4. The puzzle about temperature and prices

Montague (Montague, 1973) introduces a puzzle presented to him by Barbara Partee:

> From the premises **the temperature is ninety** and **the temperature rises**, the conclusion **ninety rises** would appear to follow by normal principles of logic; yet there are occasions on which both premises are true, but none on which the conclusion is.

By interpreting *rises* as a predicate of frames, for example, of type *AmbTemp* as given in (1) we obtain a solution to this puzzle.

(3) $\lambda r{:}\left[\text{x:}\mathit{Ind}\right]\left(\begin{bmatrix} \text{e-time} & : & \mathit{TimeInt} \\ \text{c}_{\text{tns}} & : & \text{e-time}= \iota \\ \text{c}_{\text{run}} & : & \text{rise}(r,\text{e-time}) \end{bmatrix}\right)$

Note that a crucial difference between (2) and (3) is that the first argument to the predicate 'rise' is the complete frame r rather than the value of the x field which is used for 'run'. Thus it will not follow that the value of the x field (i.e. 90 in Montague's example) is rising.

5. Fernando's string theory of events

In an important series of papers including (Fernando, 2004; Fernando, 2006; Fernando, 2008; Fernando, 2009), Fernando introduces a finite state approach to event analysis where events can be seen as strings of punctual observations corresponding to the kind of sampling we are familiar with from audio technology and digitization processing in speech recognition. (4) shows a type of event for a rise in temperature using the temperature frame *AmbTemp* in (1).

(4)
$$\left[\begin{array}{l} \text{e-time:}\mathit{TimeInt} \\ \quad \text{start:} \left[\begin{array}{l} \text{x:}\mathit{Ind} \\ \text{e-time=e-time.start:}\mathit{Time} \\ \text{e-location:}\mathit{Loc} \\ \text{c}_{\text{temp\_at\_in}}\text{:temp\_at\_in(start.e-time,} \\ \qquad\qquad \text{start.e-location, start.x)} \end{array}\right] \\ \quad \text{end:} \left[\begin{array}{l} \text{x:}\mathit{Ind} \\ \text{e-time=e-time.end:}\mathit{Time} \\ \text{e-location=start.e-location:}\mathit{Loc} \\ \text{c}_{\text{temp\_at\_in}}\text{:temp\_at\_in(end.e-time,} \\ \qquad\qquad \text{end.e-location, end.x)} \end{array}\right] \\ \text{event=start}^\frown\text{end:}\mathit{AmbTemp}^\frown\mathit{AmbTemp} \\ \text{c}_{\text{incr}}\text{:start.x}<\text{end.x} \end{array}\right]$$

6. Word meaning in flux

For all (4) is based on a very much simplified version of FrameNet's `Ambient_temperature`, it represents a quite detailed account of the lexical meaning of *rise* in respect of ambient temperature — detailed enough, in fact, to make it inappropriate for *rise* with other kinds of subject arguments. Consider price. If you look up the noun *price* in FrameNet you find that it belongs to the frame `Commerce_scenario` which includes frame elements for goods and money. If you compare the FrameNet frames `Ambient_temperature` and `Commerce_scenario`, they may not initially appear to have very much in common. However, extracting out just those frame elements or roles that are relevant for the analysis of the lexical meaning of *rise* shows a degree of correspondence. They are, nevertheless, not the same. The additional detail of the lexical semantic analysis obtained by using frames comes at a cost. *rise* appears to mean something slightly different for temperatures and prices, objects rising in location, not to mention countries as in *China rises*. We argue that there is no fixed set of meanings for a verb like *rise* but rather that speakers of a language create meanings on the fly for the purposes of interpretation in connection with a given speech (or reading) event. This idea is related to the notion of *meaning potential* discussed for example in Linell (2009) and a great deal of other literature.

7. References

Johan Bos and Malvina Nissim. 2008. Combining Discourse Representation Theory with FrameNet. In R. Rossini Favretti, editor, *Frames, Corpora, and Knowledge Representation*, pages 169–183. Bononia University Press.

Robin Cooper. 2005a. Austinian truth, attitudes and type theory. *Research on Language and Computation*, 3:333–362.

Robin Cooper. 2005b. Records and record types in semantic theory. *Journal of Logic and Computation*, 15(2):99–112.

Robin Cooper. 2010. Frames in formal semantics. In Hrafn Loftsson, Eiríkur Rögnvaldsson, and Sigrún Helgadóttir, editors, *IceTAL 2010*. Springer Verlag.

Robin Cooper. forthcoming. Type theory and semantics in flux. In Ruth Kempson, Nicholas Asher, and Tim Fernando, editors, *Handbook of the Philosophy of Science*, volume 14: Philosophy of Linguistics. Elsevier BV. General editors: Dov M. Gabbay, Paul Thagard and John Woods.

Tim Fernando. 2004. A finite-state approach to events in natural language semantics. *Journal of Logic and Computation*, 14(1):79–92.

Tim Fernando. 2006. Situations as strings. *Electronic Notes in Theoretical Computer Science*, 165:23–36.

Tim Fernando. 2008. Finite-state descriptions for temporal semantics. In Harry Bunt and Reinhart Muskens, editors, *Computing Meaning, Volume 3*, volume 83 of *Studies in Linguistics and Philosophy*, pages 347–368. Springer.

Tim Fernando. 2009. Situations in LTL as strings. *Information and Computation*, 207(10):980–999.

Charles J. Fillmore. 1982. Frame semantics. In *Linguistics in the Morning Calm*, pages 111–137. Hanshin Publishing Co., Seoul.

Jonathan Ginzburg. forthcoming. *The Interactive Stance: Meaning for Conversation*. Oxford University Press, Oxford.

Per Linell. 2009. *Rethinking Language, Mind, and World Dialogically: Interactional and contextual theories of human sense-making*. Advances in Cultural Psychology: Constructing Human Development. Information Age Publishing, Inc., Charlotte, N.C.

Richard Montague. 1973. The Proper Treatment of Quantification in Ordinary English. In Jaakko Hintikka, Julius Moravcsik, and Patrick Suppes, editors, *Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics*, pages 247–270. D. Reidel Publishing Company, Dordrecht.

Richard Montague. 1974. *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, New Haven. ed. and with an introduction by Richmond H. Thomason.

Hans Reichenbach. 1947. *Elements of Symbolic Logic*. University of California Press.

Modelling humanlike conversational behaviour

Beskow, J., Edlund, J., Gustafson, J., Heldner, M., Hjalmarsson, A. and House, D.

Department of Speech, Music and Hearing, KTH, Sweden
[beskow | edlund |jocke |mattias | annah | davidh]@speech.kth.se

Abstract

We have a visionary goal: to learn enough about human face-to-face interaction that we are able to create an artificial conversational partner that is humanlike. We take the opportunity here to present four new projects inaugurated in 2010, each adding pieces of the puzzle through a shared research focus: modelling interactional aspects of spoken face-to-face communication.

1. Introduction

Our group has a long-standing interest in humanlikeness and social signals with the visionary goal to acquire the knowledge necessary to build systems that interact more like humans do. We have a special interest in building computational models of human conversational behaviour that we evaluate in spoken dialogue systems (Edlund et al, 2008; Hjalmarsson, 2010). A prerequisite for this is of course conversational data, and we are currently running the *Spontal project* that has collected 60 hours of synchronized audio, video, and three-dimensional motion capture data in unconstrained human-human conversations, and where annotations are underway (Edlund, et al., 2010). Our current research focus represents a significant effort, we have recently initiated six new externally funded projects with a focus on describing, modelling, detecting, interpreting and synthesizing interactional aspects of spoken face-to-face communication. These projects are a continuation of our group's previous efforts in modelling and synthesizing turn-taking behaviour (Beskow et al, 2010). In research emanating from the project Vad gör tal till samtal? we explored prosodic cues in turn regulation (Edlund & Heldner, 2005). We have also investigated extralinguistic sounds such as short feedback sounds and breathing noises in turn regulation (Edlund et al, 2009b); and visual turn regulation cues in avatars as well as in systems for social interaction (Skantze & Gustafson, 2009). Throughout this work, three issues have received special attention: we stress the importance of (i) taking all available modalities into account (e.g. Edlund & Beskow, 2009); (ii) utilizing the conversational behaviour of all interlocutors and relationships formed between them to detect and interpret conversational phenomena (e.g. Edlund et al, 2009a; Neiberg & Gustafson, 2010); and (iii) the special requirements on incremental speech technology in online conversational settings (Skantze & Hjalmarsson, 2010; Skantze & Schlangen, 2009).

2. Current projects

The following is an overview of our new projects about modelling humanlike conversational behaviour.

2.1 Prosody in conversation

The project investigates how people talking to each other jointly decide who should speak when, and the role of prosody in making these joint decisions. A detailed model of the prosody involved in interaction control is crucial both for producing appropriate conversational behaviour and for understanding human conversational behaviour. Both are required in order to reach our visionary goal, and represent a artificial conversational partner. One line of inquiry within the project is the quantitative acoustic analysis of prosodic features in genuine spoken face-to-face conversations. The project focuses on local intonation patterns in the immediate vicinity of interactional events, such as transitions from (i) speech to pauses; (ii) speech to gaps; and (iii) speech by one speaker to speech by another speaker. In addition, we analyze interactional phenomena occurring on a longer time scale. In addition, the results of the acoustic analyses are fed into a second line of inquiry: studies of the effects of using such prosodic features in a conversation. These studies will include listening experiments where manipulations of genuine conversations by means of re-synthesis are used as stimuli. Furthermore, there will be pragmatic experiments where the conversational behaviour in response to the use of such prosodic features in artificial speech is analyzed. Finally, there will be analyses of conversational behaviour in response to real-time manipulations of genuine conversations.

2.2 The rhythm of conversation

The project investigates how a set of rhythmic prosodic features contributes to the joint interaction control in conversations. Of particular interest is acoustic descriptions of features related to variations in speech rate and loudness, and how these are used for interactional purposes. Loudness is generally perceived as an important component in the signalling of prosodic functions such as prominence and boundaries (cf. Lehiste & Peterson, 1959). This is highly unexplored and something we pursue in connection with rhythm as an interactional phenomenon. We want to find out, for example, whether the speech rate and loudness variations (prosodic features that are complementary to those studied in Prosody in conversation) before pauses (i.e. within-speaker silences) are different from those before gaps (between-speaker silences), or whether they display differences before backchannel-like utterances compared to other utterances.

2.3 Introducing interactional phenomena in speech synthesis

The project recreates human interactional vocal behaviour in speech synthesis in three phases. The first phase deals with what Ward (2000) calls conversational grunts like *mm* and *mhm* (Gustafson & Neiberg, 2010). We also include audible breathing, following Local & Kelly (1986) who hold breath as a strong interactional cue. These tokens are traditionally missing in speech synthesis. We remedy this by (1) annotating instances of them in the

Spontal corpus and other corpora, (2) synthesizing the missing tokens using several methods, and (3) evaluating the results in a series of experiments comparing synthesized versions with the originals as well as evaluating their perceived meaning and function. The second phase is similarly structured, but targets events that occur in the transitions between speech and silence and back –transitions that vary depending on the situation. We focus on three transition types: normal, hesitant and abrupt. In the third phase, we evaluate reactions to a dialogue system making use of the synthesized cues developed in the first two phases. In semi-automatic dialogue systems modelling speaking and listening as parallel and mutually aware processes, we use two scenarios to verify and validate our results*: the attentive speake*r – an interruptible virtual narrator making use of synthesized cues for hesitation and end-of-contribution; and *the active listener* – an information gathering system, aiming to encourage the user to continue speaking (cf. Gustafson, Heldner, & Edlund, 2008).

2.4 Intonational variation in questions in Swedish

The project investigates and describes phonetic variation of intonation in questions in spontaneous Swedish conversation, with an initial premise that there does not exist a one-to-one relationship between intonation and sentence type (Bolinger, 1989). The Spontal database is used to find a general understanding of the role of questions in dialogue and an explanation of why descriptions of question intonation has proven so difficult. We expect to find certain patterns of intonation that correlate with for example dialogue and social function.

We will test several hypotheses from the literature. One example is the hypothesis that there is a larger proportion of final rises and high pitch in questions which are social in nature than in those which are information oriented. Our results will be analyzed within the framework of biological codes for universal meanings of intonation proposed by Gussenhoven (2002). Gussenhoven describes three codes: *a frequency code* implying that a raised F0 is a marker of submissiveness or non-assertiveness and hence question intonation; *an effort code*, in which higher F0 requires increased articulation effort which highlight important focal information; and *a production code* associating high pitch with phrase beginnings and low pitch with phrase endings..

3. Summary

We have an ambitious and visionary goal for our research: to learn enough about human face-to-face interaction that we are able to create an artificial conversational partner that is humanlike in the sense that people interacting with it respond to it as they do to other humans. This visionary goal has been instrumental in the prioritization and formulation of a current research focus for our group: investigations of interactional aspects of spoken face-to-face communication. We have described four new externally funded projects that are representative of and will advance the research frontier within this common research focus. While these projects do not in themselves have either the resources or the scope to reach our visionary goal, they each add a piece of the puzzle, and we are confident that they will help identify future areas for research contributing towards the long-term goal.

4. Acknowledgments

This research is carried out at KTH Speech, Music and Hearing. Funding was provided by Riksbankens Jubileumsfond (RJ) project P09-0064:1-E *Prosody in conversation*; the Swedish Research Council (VR) projects 2009-1766 *The rhythm of conversation*, 2009-4291 *Introducing interactional phenomena in speech synthesis*; 2009-1764 *Intonational variation in questions in Swedish*; and 2006-7482 *Spontal: Multimodal database of spontaneous speech in dialog*.

5. References

Beskow, J., Carlson, R., Edlund, J., Granström, B., Heldner, M., Hjalmarsson, A., et al. (2009). Multimodal Interaction Control. In A. Waibel & R. Stiefelhagen (Eds.), Computers in the Human Interaction Loop (pp. 143-158). Berlin/Heidelberg: Springer.

Bolinger, D (1989). Intonation and its uses: Melody in grammar and discourse. London, UK: Edward Arnold.

Edlund, J., & Beskow, J. (2009). MushyPeek - a framework for online investigation of audiovisual dialogue phenomena. Language and Speech, 52(2-3).

Edlund, J., Beskow, J., Elenius, K., Hellmer, K., Strömbergsson, S., & House, D. (2010). Spontal: a Swedish spontaneous dialogue corpus of audio, video and motion capture. In Proc. of LREC 2010. Malta.

Edlund, J., Gustafson, J., Heldner, M., & Hjalmarsson, A. (2008). Towards human-like spoken dialogue systems. Speech Communication, 50(8-9), 630-645.

Edlund, J., & Heldner, M. (2005). Exploring prosody in interaction control. Phonetica, 62(2-4), 215-226.

Edlund, J., Heldner, M., & Hirschberg, J. (2009a). Pause and gap length in face-to-face interaction. In Proc. of Interspeech 2009, Brighton.

Edlund, J., Heldner, M., & Pelcé, A. (2009b). Prosodic features of very short utterances in dialogue. In M. Vainio, R. Aulanko & O. Aaltonen (Eds.), Nordic Prosody 2008 (pp. 57-68). Frankfurt am Main.

Gussenhoven, C (2002). Intonation and interpretation: phonetics and phonology. In: B Bel & I Marlien, eds, Proceedings of the Speech Prosody 2002 Conference. Aix-en-Provence, France, 47-57.

Gustafson, J., Heldner, M., & Edlund, J. (2008). Potential benefits of human-like dialogue behaviour in the call routing domain. In Perception in Multimodal Dialogue Systems, Springer.

Gustafson, J., & Neiberg, D. (2010). Prosodic cues to engagement in non-lexical response tokens in Swedish. In proc.of DiSS-LPSS.

Hjalmarsson, A (2010). Human interaction as a model for spoken dialogue system behaviour, PhD thesis, KTH, Stockholm, Sweden.

Lehiste, I, & Peterson, G E (1959). Vowel amplitude and phonemic stress in American English. JASA, 31: 428-435.

Local, J K, & Kelly, J (1986). Projection and 'silences': Notes on phonetic and conversational structure. Human Studies, 9: 185-204.

Neiberg, D., and Gustafson, J. (2010). "Modeling Conversational Interaction Using Coupled Markov Chains", In Proceedings of the DiSS-LPSS Joint Workshop 2010.

Skantze, G., & Gustafson, J. (2009). Attention and Interaction Control in a Human-Human-Computer Dialogue Setting. In Proc SigDial 2009.

Skantze, G., & Hjalmarsson, A. (2010). Towards Incremental Speech Production in Dialogue Systems. In proc. of SigDial. Tokyo, Japan.

Skantze, G., & Schlangen, D. (2009). Incremental dialogue processing in a micro-domain. In Proc EACL-09. Athens, Greece.

Ward, N (2000). The challenge of non-lexical speech sounds. In Proceedings of ICSLP 2000. Beijing, China, 571-574.

GRASP: Grammar-based Language Learning

Peter Ljunglöf

DART: Centre for Augmentative and Alternative Communication (AAC) and Assistive Technology (AT)
and Språkbanken, Department of Swedish Language, University of Gothenburg
peter.ljunglof@gu.se

Abstract

We are developing a pedagogical tool to support language learning and training for children with communicative disabilities. The system has a graphical interface, where the user can move, replace, add, and in other ways modify, words or phrases. The system keeps the sentence grammatical, by automatically rearranging the words and changing inflection, if necessary. In this way we hope that the system stimulates the child to explore the possibilities of language.

1. Introduction

In the GRASP[1] project, financed by Sunnerdahls Handikappfond, we are developing an interactive system for Computer Assisted Language Learning (CALL) (Davies, 2010). There are two intended target groups: one is children and adults trying to learn another language; another group is persons with communicative disabilities who are learning to read and write in their first language.

The idea is that it will work as an interactive textbook, where the user can read different texts (just as in a traditional textbook) but also experiment with and modify the texts. The system will be divided into modules dealing with different linguistic features, e.g., inflection, word classes, simple phrases and more advanced constructions. The modules can be used on their own, or can be combined for more advanced training.

The texts are stored in an internal grammar format which makes it possible to transform sentences interactively, while still keeping them grammatical. The underlying grammar is multilingual, which is useful not only for second language learning, but also for first language learning for persons with communicative disorders, since words and phrases can be interpreted in a symbol language such as Blissymbolics.

The system has a graphical user interface, where each word acts a kind of icon that can be clicked, moved, replaced, or modified in other ways. When the user moves a word to a new position, or changes the inflection of a word, the system automatically rearranges the other words and changes inflection so that the sentence stays grammatically correct.

2. System description

In this section we describe the final GRASP system, which is currently under development. Note that all features are *not* currently implemented (as of August 2010).

As the basic component we are using Grammatical Framework (GF) (Ranta, 2009b), a modular and multilingual grammar formalism. On top of this we build the graphical interface which the user interacts with. As a glue between the grammar and the interface, we implement an API for modifying syntax trees using linear constraints and a tree similarity measure.

2.1 Ready-made texts

The system will contain a number of texts that the user can read and experiment with. The texts are stored as GF grammars which makes them possible to modify in a grammatical way. Since GF is multilingual, the texts can be linearized in parallel for several languages. This can be useful for second language learning, as the system can display the text in the user's first language in parallel. Multilinguality is also useful for first language learning, e.g., by displaying the parallel text in a symbol language such as Blissymbolics.

2.2 Graphical interaction

The words in the example texts are icon-like objects which can be clicked on, moved around and deleted. If the user clicks on a word, a context menu appears consisting of similar words, such as different inflection forms, or synonyms, homonyms, etc. When a new word form is selected from the menu, it replaces the old word, and if necessary, the nearby words are also modified and rearranged to keep the sentence grammatical.

The user can move a word to another position in the sentence, and the system will automatically keep the sentence grammatical by rearranging and change inflection, if necessary. Words can be deleted from the sentence by dragging them away. The user can also add or replace words by dragging new words into the sentence. All the time, the sentence will adapt by rearranging and changing inflection.

The system can also be used for exercises and tests, by turning off the automatic rearrangement and instead show problematic phrases in another colour. One example exercise could be to turn a given sentence into passive form by moving words and changing their inflection until the sentence is correct. Multlinguality can also be used for exercises, e.g., to build a correct translation of a sentence by moving and modifying the translated words.

2.3 Grammar modules

Different grammatical and linguistic constructions are put in separate grammar modules, which the user him/herself can choose to train. Several modules can be chosen at the same time, for training combined phrases. Examples of

[1] GRASP is an acronym for "grammatikbaserad språkinlärning" (grammar-based language learning).

constructions that can be put into modules of their own are prepositional phrases, relative clauses, adjectives, passive form, word compounds, topicalization, conjunctions, and infinitive phrases.

2.4 No free text input

The system does not allow the user to enter words, phrases or sentences from the keyboard. There are several reasons for this, but the main reason is to avoid problems with words and grammatical constructions that the system doesn't know anything about. Systems that are supposed to handle free text input sooner or later run into problems with unknown words or phrases (Heift, 2001).

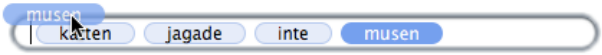
3. An illustrative example

As an explanatory example, we show how to transform a sentence in active form (*katten jagade inte musen – the cat didn't chase the mouse*) into passive form (*musen jagades inte av katten – the mouse wasn't chased by the cat*), in two different ways.

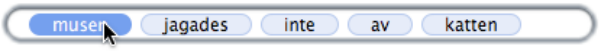
3.1 Moving a word to another position

We start by grabbing a word, in this case the word "musen" which is in object position:

While we move the word the sentence remains unaffected, but the marker gives a hint of where the word can be inserted:



Finally we drop the word in its new subject position, but the resulting sentence (*musen katten jagade inte*) is not correct. Therefore the system rearranges the sentence to the closest possible grammatical. In this case the sentence is transformed into passive form:

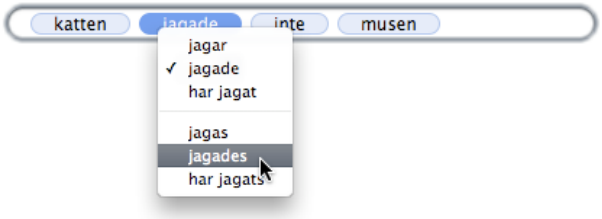


If a topicalization module had been active instead of a passive form module, the system would have topicalized the sentence (*det var musen som katten inte jagade – it was the mouse that the cat didn't chase*).

What will not happen is that the mouse becomes the subject instead of the cat (*musen jagade inte katten*), since it involves two changes in the GF syntax tree (changing the subject and changing the object), whereas passive form or topicalization only involves one change.

3.2 Choosing verbform in the context menu

Another way of turning the sentence into passive form is to select from the context menu of the verb:



Note that the contents of the context menu depends on which grammar module is active. If the topicalization module had been active, the word "musen" would get its context menu extended with "det var musen" or something similar.

4. Implementation

The system consists of three implementation layers. The bottom layer is the GF grammar formalism (Ranta, 2009b). We use GF's multilingual resource grammar (Ranta, 2009a) to define the different grammar modules. The example texts are stored as GF syntax trees, and the GF linearization algorithm is used for displaying the sentences to the user. We have no use of parsing the sentences, since the syntax trees are already known and there is no free text input.

On top of GF we have implemented an API for modifying syntax trees by specifying linearization constraints. The API consists of functions that transform trees to obey the constraints, by using as few transformations as possible. An example of constraints can be that the linearizations of some given tree nodes must come in a certain order (e.g., when the user moves a word to a position between two other words). Another example is that the linearization of a given node must be of a specified form (e.g., when the user select a specific word form from the context menu).

For the API functions to work, we have defined a similarity measure between GF trees. This is based on the notion of *tree edit distance* (Bille, 2005), but with modifications to ensure type-correctness according to the GF type system.

The final layer is the graphical interface, which communicates with the API to decide which words can be moved where, and what their context menus should contain.

5. Discussion

The GRASP system is work in progress, and not all features described in section 2 are implemented:

The grammar is a monolingual Swedish grammar, and the module system is not fully developed yet. The grammar curently handles noun phrase inflection, fronting of noun phrases, and verb inflection. The graphical interface cannot yet handle all kinds of interaction, only context-click and movement; the underlying API however is more mature.

Our plan is to have a working demonstration system by the end of 2010.

6. References

Philip Bille. 2005. A survey on tree edit distance and related problems. *Theoretical Computer Science*, 337(1–3):217–239.

Graham Davies. 2010. Information and Communications Technology for Language Teachers (ICT4LT). Accessed 26 aug 2010 from `http://www.ict4lt.org/en/`.

Trude Heift. 2001. Intelligent language tutoring systems for grammar practice. *Zeitschrift für Interkulturellen Fremdsprachenunterricht*, 6(2).

Aarne Ranta. 2009a. The GF resource grammar library. *Linguistic Issues in Language Technology*, 2.

Aarne Ranta. 2009b. Grammatical Framework: A multilingual grammar formalism. *Language and Linguistics Compass*, 3(5):1242–1265.

Using the pyramid method to create gold standards for evaluation of extraction based text summarization techniques

Bertil Carlsson, Arne Jönsson

Department of Computer and Information Science, Santa Anna IT Research Institute AB
Linköping University, SE-581 83, Linköping, SWEDEN
Berca955@student.liu.se, arnjo@ida.liu.se

Abstract

We present results from using a version of the pyramid method to create gold standards for evaluation of automatic text summarization techniques in the domain of governmental texts. Our results show that the pyramid method may be useful to create gold standards for extraction based summarization techniques using only five human summarisers.

1. Introduction

Automatic creation of text summarizations is an area that has gained an increasing interest over the years, for instance in order to allow for skim reading of texts or to facilitate the process of deciding if a text is interesting to read in full. In order to know if the summarization is useful it must be evaluated.

To evaluate automatic text summarization techniques we either need humans to read, and evaluate a number of summarizations, or we can compare it to a gold standard, a "correct" summarization of the text, i.e. extrinsic or intrinsic evaluation of the text. A gold standard is often a compilation of different human created summarizations which is then put together into one.

It is an open question how to assemble such human created summaries into one gold standard. In this paper we present results from using a variant of the pyramid method (Nenkova, 2006) to create gold standards of text summaries. We use the pyramid method on extraction based summaries, i.e. we do not ask our human summarisers to write an abstract summary but to extract a number of whole sentences from the text. The texts are governmental texts. We also present an evaluation of our gold standards.

2. The pyramid method

The pyramid method is a summarization technique used to assemble summary fragments (words, phrases or sentences) from different humans to generate one summarization (Nenkova, 2006). Nenkova used information fragments, brief phrases with the same information content, in her original study in the domain of news texts.

The pyramid method assigns each information fragment a weight, reflected by the number of human summarisers that have highlighted it as an important fragment for the text. Each fragment is then inserted into a pyramid where each layer in the pyramid represents how many summarisers that have suggested the fragment. Consequently, the number of layers in the pyramid is equal to the number of summarisers and the higher up the more likely it is that a fragment is important.

One interesting result from Nenkova (2006) is that pyramids comprising four to five layers produce the best results in evaluations of summaries. Thus, contrary to e.g. Halteren and Teufel (2003), five summaries is all that is needed to produce a gold standard.

3. Creation of the gold standards

We use 5 frequently used fact sheets from the Swedish Social Insurance Administration (Sw. Försäkringskassan) as selected by employees at the Swedish Social Insurance Administration. They comprise 62-91 sentences, each between 1000 and 1300 words. All texts were about allowances and had the same structure.

Our ambition was to create indicative summaries, i.e. they should not replace reading the whole text but rather facilitate deciding if reading the whole text is interesting. A pre-study revealed that 10% is an appropriate length of such a summary (Jönsson et al., 2008).

Five persons created summaries of all five texts, two students, two seniors and one worked in a private company. All had sufficient read and write skills in Swedish and none had ever constructed extraction based summaries before.

The text summarizations were entered into a pyramid, as explained in Section 2., one for each text, and from these the gold standards were created. The variation between the summaries produced by the summarisers versus the produced gold standard were investigated by computing the sentence overlaps for the summaries.

The sentence overlap for the five gold standards created in this study varies between 57,5% and 76,6%, which is in line with previous studies that have found that the sentence overlap normally vary between 61% and 73% where the larger number is achieved by more accustomed summarisers (Hassel and Dalianis, 2005). All but one of the summaries obtain the minimum value which represents a good overlap according to (Hassel and Dalianis, 2005). The 57,5% overlap can be explained by inexperience from the human summarisers part in creating extraction based summaries. Something which has been well documented in earlier work, such as Hassel and Dalianis (2005).

To further investigate the variation obtained by our human summarisers, we calculated the number of new sentences added by each human summariser. These investigations show that the number of new sentences added by the summarisers drops rather quickly. At most the fifth summariser adds three new sentences and at best only one. Thus, we can assume that the summaries comprise the most important sentences from the text. It should be noted that

humans do not agree on what is a good summary of a text (Lin and Hovy, 2002; Hassel and Dalianis, 2005; Jing et al., 1998), which means that there is probably not one single best summary. The results presented here also point towards texts having a limit on important sentences that should be included in summaries. Something that has to be further investigated.

4. Evaluation

Evaluation of the gold standards was conducted by having subjects read the summaries and answer a questionnaire on the quality of the summary. The questionnaires used six-point Likert items and comprised the following items on the summary: [Q1] ... has a good length to give an idea on the content in the original text, [Q2] ... is experienced to be information rich, [Q3] ... is experienced as strenuous to read, [Q4] ... gives a good idea on what is written in the original document, [Q5] ... gives a good understanding of the content of the original document. [Q6] ... is experienced as missing relevant information from the original document, and [Q7] ... is experienced as a good complement to the original document.

The subjects for our evaluation where 10 students and 6 professional administrators at the Swedish Social Insurance Administration.

All subjects read the summary but did not have the original text at hand, to more resemble future use of the system. Discourse coherence for extraction based summaries is, of course, a problem. Our evaluators were not instructed to disregard discourse coherence since this is a factor which has to be accounted for when creating texts of this sort.

The results from the student evaluations are presented in Table 1. Note that, as the items are stated, a high score is considered positive on Q1, Q2, Q4, Q5 and Q7 whereas as low score on Q3 and Q6 is considered positive. Note also that the questions themselves are intertwined and hence act as some sort of control questions to each other in order to assure that the data given by the participants in the questionnaire is correct.

Table 1: Mean from the students' responses

	Q1	Q2	Q3	Q4	Q5	Q6	Q7
S1	4,5	4,5	2,8	4,0	3,8	2,5	4,2
S2	4,7	4,8	1,5	4,2	4,6	2,2	4,5
S3	5,2	5,1	2,0	4,4	4,6	1,9	4,7
S4	4,9	5,3	2,2	4,7	4,9	2,1	4,7
S5	4,5	4,2	1,9	4,3	4,4	2,8	4,5

As can be noted from Table 1 the evaluators give positive opinions on all items.

Table 2: Mean from the professionals' responses

	Q1	Q2	Q3	Q4	Q5	Q6	Q7
S1	4,0	4,2	4,0	4,2	4,2	2,5	4,2
S2	4,7	4,5	2,8	4,3	4,2	2,3	4,3
S3	4,5	4,5	3,0	4,5	4,7	2.2	4,8
S4	4,5	4,7	2,2	4,7	4,7	1,7	5,0
S5	4,5	4,0	3,5	4,3	4,5	1,8	4,0

The results from the professional administrators' answers to the questionnaires, Table 2, also demonstrate positive opinions on all items, but Q3. The professional administrators are indifferent regarding how hard the texts are to read. In fact, two subjects rank them as rather hard to read.

Notable is that the students and professional administrators provide very similar answers to most of the questionnaires. They all consider the text to be informative, Q2, and having an appropriate length, Q1. They also, all think that the texts provide a good idea on what was in the original text, Q4 and Q5. Furthermore, the subjects do not think that the texts miss relevant information.

5. Summary

We have used the pyramid method to create extraction based summaries of governmental texts.The summaries are evaluated by both novices (students) and professionals (administrators at the local governmental agency) and the evaluations show that the summaries are informative and easy to read.

Our results are in line with previous research (Nenkova, 2006) which states that five human summarisers are enough to produce a gold standard. It can be further stated that the pyramid method then not only can be used in order to create gold standards from abstract summaries but also from extraction based summaries.

Acknowledgements

This research is financed by Santa Anna IT Research Institute AB. We are grateful to our evaluators and especially the staff at the Swedish Social Insurance Administration.

6. References

Hans Van Halteren and Simone Teufel. 2003. Examining the consensus between human summaries: Initial. In *In HLT-NAACL DUC Workshop*, pages 57–64.

Martin Hassel and Hercules Dalianis. 2005. Generation of Reference Summaries. In *Proceedings of 2nd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, Poznan, Poland, April 21-23.

H Jing, R Barzilay, K McKeown, and M Elhadad. 1998. Summarization evaluation methods: Experiments and analysis. *AAAI Symposium on Intelligent Summarization*, Jan.

Arne Jönsson, Mimi Axelsson, Erica Bergenholm, Bertil Carlsson, Gro Dahlbom, Pär Gustavsson, Jonas Rybing, and Christian Smith. 2008. Skim reading of audio information. In *Proceedings of the The second Swedish Language Technology Conference (SLTC-08), Stockholm, Sweden*.

Chin-Yew Lin and Eduard Hovy. 2002. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 45–51, Morristown, NJ, USA. Association for Computational Linguistics.

Ani Nenkova. 2006. *Understanding the process of multi-document summarization: Content selection, rewriting and evaluation*. Ph.D. thesis, DigitalCommons@Columbia, January 01.

Directing conversation using the prosody of mm and mhm

Joakim Gustafson and Daniel Neiberg

Department of Speech, Music and Hearing, KTH, Sweden
jocke@speech.kth.se, neiberg@speech.kth.se

Abstract

This paper investigates the prosodic patterns of conversational grunts in a Swedish letter-to-the-editor call-in radio show. The feedback of a professional speaker was investigated to give insight in how to build a simulated active listener that could encourage its users to continue talking. Possible domains for such systems include customer care and second language learning. The prosodic analysis of the conversational grunts showed that the perceived engagement level decreases over time.

1. Introduction

Today's spoken dialogue systems are being considered for applications such as social and collaborative applications, education and entertainment. These new areas call for systems to be increasingly human-like in their conversational behaviour (Hjalmarsson, 2010). In human-human conversations both parties continuously and simultaneously contribute actively and interactively to the conversation. Listeners actively contribute by providing feedback such as conversational grunts. Their feedback indicates attention, feelings and understanding, and its purpose is to support the interaction (Yngve, 1970). According to Ward (1998) the important prosodic features of conversational grunts are: loudness, height and slope of pitch, duration, syllabification, duration and abruptness of the ending. These features were used in a study on the prosody of acknowledgements and backchannels in task oriented dialogues (Benus et al. 2007).

In order to develop systems that can achieve the responsiveness and flexibility found in human-human interaction, it is essential that they process information incrementally and continuously rather than in turn sized chunks. Conversational grunts, audible breathing and self-corrections are abundant in conversational speech. We have recently initiated a three-year research project that aims at adding human interactional verbal behavior in speech synthesis. This paper investigates the prosody of Swedish conversational grunts.

2. The attentive listener database

In the current study we have analysed response tokens in a corpus of 73 calls to a Swedish phone-in radio program. The program is called Ring P1, and it allows members of the public to call in and share their opinions on current affairs. We have selected six 45-minute programs hosted by Täppas Fogelberg. In this study we have selected the main phases of 73 calls - a dialogue corpus of about three hours. During theses main phase the callers produced on average 22 inter pausal units (IPUs) that in half of the cases were followed by speaker shifts. In about 80% of these speaker shifts the radio host merely produced short backchannel continuers that encouraged the caller to continue speaking. This means that the radio host mostly acted as an active listener.

2.1 Data selection and tagging

Since the recordings of the radio programs are recorded in mono the first step was to manually annotate the speech for speaker, (where overlapped speech was labelled as both). The syllable boundaries of the last three syllables of the caller IPUs were manually assigned. The response tokens were tagged as lexical (e.g. "ja") or non-lexical (e.g. "mm") and as monosyllabic ("mm") or bisyllabic ("mhm"). In the 73 dialogues there were 174 lexical and 459 non-lexical response tokens, out of which 44% were perceived as bisyllabic. In this study the prosodic patterns of the non-lexical response tokens "mm" and "mhm" have been investigated. For these conversational grunts pitch contour, intensity distribution and syllable boundaries were manually labelled. In Table 1 the appearance of the most common prosodic contours are visualized in pitch curves where the line width indicate the intensity.

	early	late	two peaks	even
fall				
rise				
fall-rise				

Table 1. Examples of intensity modulated pitch curves, with pitch movement in rows and intensity distribution in columns.

All response token were also labelled for engagement level, where passive corresponds to acknowledgement that the radio host is still listening, while active response tokens signal interest and encourages the caller to say something more. The pitch slope of the loudest part of the pitch curves of the bisyllabic tokens correlates closely to the perceived engagement of the feedback: 90% of the bisyllabic response tokens that had a falling pitch on the loudest syllable were perceived as unengaged, while 80% of the tokens with rising pitch on the prominent syllable were perceived as engaged. In bisyllabic response tokens with two intensity peaks or even intensity there was a 50/50 split in the engagement ratings.

3. Signal processing

We use the ESPS pitch tracker and logarithmic power function in the SNACK toolkit with a 10ms frame rate. The F0 values are then converted to semitones. Any unvoiced frames between voiced frames are interpolated over using splines. Then a median filter with a 3 frame window is applied, followed by a moving average filter with a 5 frame window. This filtering procedure is applied to both the intensity and to pitch. Each feedback is assigned a parameter x which is the elapsed time from the start off the dialog divided by the total dialog duration.

This study suggests a data-driven intonation model based on a modified length invariant cosine transform (DCT). Each contour f(n) with N points is parameterized by

$$c(k) = \frac{1}{N}\sum_{n=1}^{N} f(n)\cos\left(\frac{\pi}{N}\left(n-\frac{1}{2}\right)(k-1)\right) \quad (1)$$

Both the pitch and intensity contours can effectively be parameterized by a few coefficients with this method. We want to find prototypical contours as a function of x. To do this an automatic clustering method is used: Initially, one feature vector per feedback is constructed by using the first K DCT coefficients for F0 and intensity. The feedback length is also added to the vect. We use K = 3 for monosyllabic and K = 5 for bisyllabic tokens. Then vector quantization is performed by sweeping x in steps of 0.05. The number of clusters is chosen such that all significant minima in average distortion are found. The centroids (mean values) are transformed using inverse DCT, stretched to the average duration and plotted in Figure 1.

4. Conclusions

In this study we have investigated the prosodic patterns of conversational grunts in a Swedish call-in radio show. The professional active listener mostly responded with response tokens at pauses in the callers' speech. Grunts with a rising pitch are associated with interest and encouragement for more speech from the interlocutor, and those with falling pitch function as acknowledgement and signals lesser interest. For bisyllabic response tokens it is the pitch slope of the loudest syllable that decides which of these two engagement levels the grunt signals. The distribution of grunts with different pitch contours changes as a function of dialogue position. The interest-signaling and encouraging pitch contours are most common at the beginning of the call. Over time the mean intensity of the feedbacks decreases, the bisyllabic becomes flatter and the overall pitch level decreases. At the very end this pattern changes where the mean and slope of the pitch increases slightly. The implication of our results on conversational speech synthesis is that if we want to synthesize conversational grunts it is not enough to add the sounds of conversational grunts like "mhm" and control the pitch and duration. In order to display the different functions and degrees of interest we also need to be able to control the intensity level continuously on the individual syllables.

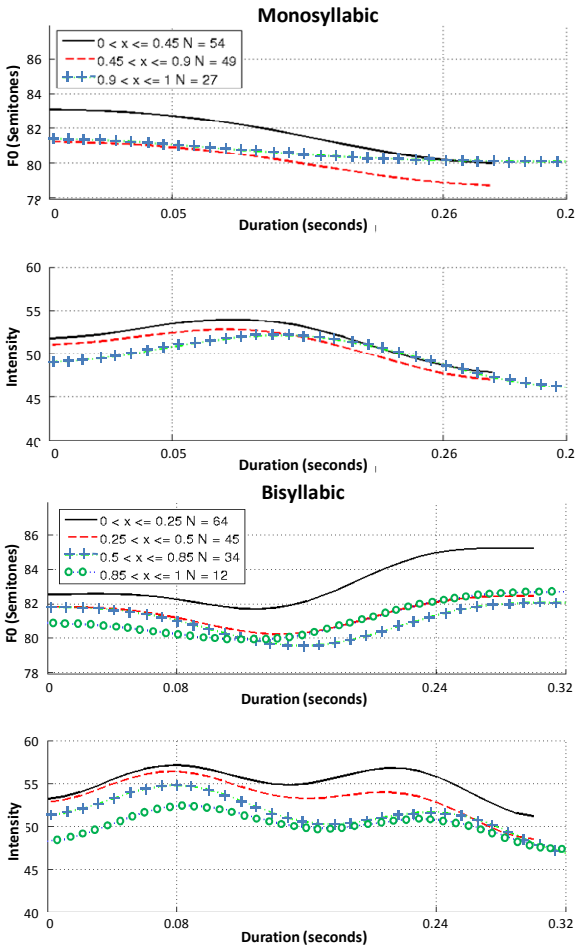


Figure 1: Pitch and intensity curves as a function of the relative position in dialog. Monosyllabic feedback at the top and bisyllabic at the bottom.

5. Acknowledgments

This research is carried out at KTH Speech, Music and Hearing. Funding was provided by the Swedish Research Council (VR) project "Introducing interactional phenomena in speech synthesis" (2009-4291).

6. References

Benus, S., Gravano, A., & Hirschberg, J. (2007). The prosody of backchannels in American English. In *Proceedings of ICPhS XVI* (pp. 1065-1068). Saarbrücken, Germany.

Hjalmarsson, A (2010). Human interaction as a model for spoken dialogue system behaviour, PhD thesis, KTH, Stockholm, Sweden.

Ward, N. (1998). The Relationship between Sound and Meaning in Japanese Back-channel Grunts. In *4th Meeting of the (Japanese) Association for Natural Language Processing*.

Yngve, V. H. (1970). On getting a word in edgewise. In *Papers from the sixth regional meeting of the Chicago Linguistic Society* (pp. 567-578). Chicago.

Towards a Rule Based System for Automatic Simplification of Texts

Jonas Rybing, Christian Smith
Linköping University

Annika Silvervarg
Department of Computer and Information Science
Linköping University
annika.silvervarg@liu.se

1 Introduction

The need for simplified texts in various areas is increasing, however manually simplification of texts is very resource intense and costly. An automatic system for simplification of texts is therefore very desirable. In this paper we present the initial development of such a system for Swedish and discuss results from an evaluation based on various mathematical measures for simplified texts.

2 The CogFLUX system

The CogFLUX system is based on transformation rules used to reduce complexity of texts. The rules were compiled by Anna Decker (2003) based on studies of corpora of easy to read texts and normal texts. She has identified 25 general transformation rules used to simplify a text syntactically. The rules can be grouped into two subsets of rules; 1) rules that remove or replace sub phrases and 2) rules that add new syntactical information to the text. An example of a rule from the first category is: np(det+ap+n) → np(n). This rule will replace any nominal phrase containing a determiner, an adjective phrase and a noun with a nominal phrase containing only the noun. CogFLUX implements the first subset of Decker's rules. The system also replaces abbreviations with its extended form based on list of abbreviations assembled by the Swedish Academy.

3 Evaluation measures

The formulas used in this study are Swedish readability index (LIX), noun quota (NQ) and lexical variation (OVIX). These are all mathematical formulas resulting in a strict quantitative value. The advantage of using quantitative measures is that they can be applied automatically and the results are easy to compare.

The measure LIX, developed by Björnsson (1968), has been extensively used to measure the readability of Swedish texts. LIX is calculated using the formula:

$$LIX = \frac{O}{P} + 100\frac{L}{O}$$

where O is the number of words in a text, P is the number of sentences in a text and L is the number of long words, i.e. words with more than 6 characters.

Measuring the amount of information in a text can be done with the NQ measure. A result around 100 is regarded as a normal ratio of information representing that of newspapers (Josephson et al., 1990). The information ratio is calculated by:

$$NQ = \frac{Number\ of\ nouns}{Numbers\ of\ Verbs} \times 100$$

OVIX is a ratio measure the number of unique words in the text, representing how rich of a variation of words used in the text. A high value, i.e. rich text, is associated with lower readability (Lundberg & Reichenber, 2008). OVIX is calculated by:

$$OVIX = \frac{Number\ of\ uniqe\ words}{Number\ of\ total\ words}$$

4 Results

The evaluation material used was a collection of texts with a total of 100 000 words, of which 50% was fiction, 25% was newspaper articles and 25% was public authority documents. These were automatically simplified by CogFLUX and the three evaluation measures were then computed for the resulting simplified texts. Seven sets of transformation rules were used in the evaluation. The sets were composed and categorized by what type of phrase the rules manipulated, adjective phrases (AP), noun phrases (NP) or preposition phrases (PP). Some of the sets are combinations of different rules, e.g. a set with rules manipulating both noun and preposition phrases (NP+PP). In table 1 the results of the evaluation are presented. The first column displays the text categories. The second column displays the type of measure and the remaining columns shows which rule set was applied and their resulting value. For comparison, manually written easy to read texts accumulated by Katarina Mühlenbock at the University of Gothenburg is also included in the last column (Manual). The texts in this corpus are distributed accordingly to the distribution of the texts used in this study, but this corpus consisted of about one million words.

The LIX value actually increases slightly for all of the texts regardless of applied rule set. The biggest increase can be found in where the prepositional (PP) and to some extent the noun (NP) phrase rules where applied. When phrases are deleted it will only change the LIX positively if the phrase contained a majority

		Rule sets								
		All	NP+ PP	NP+ AP	PP+ AP	PP	AP	NP	No	Manual
Fictive texts	LIX	44	46	42	43	46	42	45	41	24
	NQ	46	46	66	46	46	66	66	66	55
	OVIX	14	15	13	14	15	13	14	14	0,07
News-paper articles	LIX	56	59	53	56	58	53	56	52	36
	NQ	88	88	126	88	88	126	126	126	123
	OVIX	25	25	22	24	25	21	22	22	32
Authority documents	LIX	54	56	51	51	56	51	54	51	35
	NQ	90	90	122	122	90	122	122	122	90
	OVIX	7	7	6	6	7	6	7	6	1,28
All	LIX	49	52	47	49	51	47	50	46	29
	NQ	64	64	90	64	64	90	90	90	75
	OVIX	12	12	11	12	12	10	11	11	-

Table 1 Evaluation measures for different rule sets and different text genres, as well as manually simplified texts.

of long words, and since prepositional phrases often are composed of short words, the deletion of them affects the LIX negatively. Another reason for the higher LIX values is that a guideline for easy to read texts, applied in CogFLUX, is to replace abbreviations with the full form. When this occurs a short abbreviation is exchanged with one or more words, long or short, but the total number of sentences remains unchanged, which cause increase in the LIX value.

The measures NQ values drops noticeably when PP rules where applied. In regards to the normal NQ value of 100 the measured 46 for fictional texts is a very low value. This indicates that there is a lower number of nouns, prepositions and participle per word in the texts after the performed simplification.

The OVIX value tends to drop somewhat when the AP rules are applied and increase slightly when the PP rules are used. It therefore seem as adjective phrases contain rarely used words and prepositional phrases contain frequently used words.

The sets of rules used by CogFLUX are manually induced based on ly newspaper articles. There was no observed difference in performance between the different genres. This imply that the rules although induced from one type of texts are general, at least in the aspect of making the same errors and same correct simplifications between the genres.

The values of LIX are considerably lower for the manually generated texts than the values of their automatically generated counterpart. The OVIX values are also lower which can partially be explained by the difference in size of the corpora. The ratio of unique word per words will inevitably drop when a corpus grows bigger. The NQ value is overall lower for the automatically generated texts with the exception of the public authority documents.

5 Discussion

The measurements used should only be seen as indications, with easy to read texts correlating with low values. It was clear that they are not enough to fully determine the readability of a text, as the text often seemed to lose coherence with fragmented sentences despite getting better results on the measures. This indicates that the measurements should be complimented with some way to measure readability on a more grammatical level, the coherence of the whole text, or the relevancy of information kept or deleted.

As of today, CogFLUX accepts all suggestions generated by Decker's transformation rules and performs them accordingly. However, Decker found that there are times when transformation rules not is applicable, thus the transformations should not always be performed. Because of this, the transformations are more often than not performed at the wrong place at the wrong time, effectively deleting important information and resulting in a fragmented text. Thus, the simplification rules are not enough to simplify texts on their own. The system need some way, using decision making or further heuristic, of determining when to apply a rule and when not to.

Reference

C. H. Björnsson. Läsbarhet. Bokförlaget Liber AB, 1968.

A. Decker. Towards automatic grammatical simplification of swedish text. Master's thesis, Stockholm's University, 2003.

O. Josephson, L. Melin, and T. Oliv. Elevtext. Analyser av skoluppsatser från åk 1 till åk 9. Lund: Studentlitteratur, 1990.

I. Lundberg and M. Reichenberg. Vad är lättläst? 2008.

Constructing a Swedish General Purpose Polarity Lexicon Random Walks in the People's Dictionary of Synonyms

Magnus Rosell, Viggo Kann

KTH CSC
100 44 Stockholm, Sweden
rosell@csc.kth.se, viggo@csc.kth.se

1. Introduction

In *opinion mining* or *sentiment analysis* one task is to assign polarity to a text or a segment (Pang and Lee, 2008). Methods for this can be helped by lexical resources with polarity assigned to words and/or phrase. We aim to construct a large free Swedish general purpose polarity lexicon.

There are many available polarity resources for English and several descriptions of how to create them, see Pang and Lee (2008). Many such methods use some other lexical resource, such as thesauruses and lexicons, viewed as a graph of word relatedness. Hassan and Radev (2010) use random walks on such a graph and achieve better and comparable results to previous work. Random walks on the graph consider all paths between two words, as opposed to only the shortest.

Velikovich et. al. (2010) derive a large polarity lexicon from web-documents, which is not limited to specific word classes and contains slang and multi-word expressions. They find that it gives better performance in sentence polarity classification than lexicons constructed from ordinary lexical resources such as WordNet.

2. The People's Dictionary of Synonyms

The People's Dictionary of Synonyms (Kann and Rosell, 2005) contains words from different stylistic classes, both slang and formal words appear. It also does not distinguish between different word classes. Synonymity is defined by the users.

The dictionary was constructed in two steps. In the first a list of possible synonyms was created by translating all Swedish words in a Swedish-English dictionary to English and then back again using an English-Swedish dictionary. The generated pairs contained lots of non-synonyms. The worst pairs were automatically removed using Random Indexing.

In the second step every user of the popular dictionary Lexin on-line was given a randomly chosen pair from the list, and asked to judge it. An example (translated from Swedish): "Are 'spread' and 'lengthen' synonyms? Answer using a scale from 0 to 5 where 0 means I don't agree and 5 means I do fully agree, or answer I do not know." Users could also propose pairs of synonyms, which subsequently were presented to other users for judgment.

All responses were analyzed and screened for spam. The good pairs were compiled into the dictionary. Millions of contributions have resulted in a constantly growing dictionary of more than 80 000 Swedish pairs of synonyms. Since it is constructed in a giant cooperative project, the dictionary is a free downloadable language resource.

Each synonym pair in the dictionary has a grade. It is the mean grading by the users who have judged the pair. The available list contains 16 006 words with 18 920 pairs that have a grading of 3.0 to 5.0 in increments of 0.1. The dictionary can be considered an undirected weighted graph. It has 2 268 connected components, the second largest of which consists of 35 words and 46 pairs. In the following work we only use use the largest component, which we call Synlex. It consists of 9 850 words and 14 801 pairs.

3. Method

We use a method very similar to Hassan and Radev (2010). However, in Synlex we have weights on the edges, a measure of relatedness, which we exploit.

Synlex is a graph $G = (V, E)$, where $V = \{i\}_{i\in[1,\dots n]}$ is the set of n words, and $E = \{(i, j)\}_{i,j\in V}$ is the set of edges or links between the words, corresponding to the synonym pairs of Synlex. With each edge in E we associate three values. First, the synonymity level of Synlex: $\mathrm{syn}(i, j) \in [3.0, 5.0]$. We define the length of an edge as $\mathrm{len}(i, j) = 5.0/\mathrm{syn}(i, j)$, i.e. we consider words with high synonymity to be close to each other. Finally, we define the transition probability associated with each edge:

$$\mathrm{prob}(s, d) = \frac{\mathrm{syn}(s, d)}{\sum_{(s,j)\in E} \mathrm{syn}(s, j)}. \quad (1)$$

Thus the random walk we use takes the synonymity level of Synlex into account in deciding to which node to go next, and the length of each edge.

See Figure 1 for the random walk method. We have used $I = 100$ and $m = 250$ and the following seedwords:

- positive: S_+ = *{ positiv, bra, glad, rolig }*
- negative: S_- = *{ negativ, dålig, ledsen, tråkig }*

The random walk may result in different values everytime. To study this we repeat the method 10 times for each word and calculate mean values and standard deviations.

4. Results and Discussion

In Table 1 we give som examples of words with their polarity values after applying the method to Synlex. We present the words that were deemed most positve and negative, as well as some of those deemed neutral, and some further positve and negative examples.

Most Positive		Neutral		Most Negative		More Examples	
Word	Value	Word	Value	Word	Value	Word	Value
på bra humör	252.3 (0.3)	...		...		duktig	24.6 (9.6)
inte dåligt	232.2 (0.2)	envig	0.0 (0.1)	krasslig	−64.8 (27.8)	godtagbart	24.3 (7.9)
positivt	207.0 (0.1)	skrammel	0.0 (0.1)	låg	−75.4 (18.7)	euforisk	24.2 (8.7)
fryntlig	201.9 (0.1)	fortbestå	0.0 (0.2)	tristess	−78.3 (15.0)	säll	23.8 (10.1)
på gott humör	191.8 (0.2)	krasch	0.0 (0.2)	tradig	−82.8 (26.8)	läckert	23.3 (7.1)
på topp	181.7 (0.1)	bestraffning	0.0 (0.1)	grå	−87.0 (18.9)	superbra	22.5 (11.1)
suveränt	166.5 (0.1)	uppsikt	0.0 (0.1)	sårad	−101.3 (22.2)	sprallig	20.9 (9.3)
gladsint	156.5 (0.1)	tillskott	0.0 (0.1)	suger	−112.4 (28.4)	kalas	20.7 (19.8)
uppåt	156.5 (0.1)	fekalier	0.0 (0.1)	illa	−113.9 (29.0)	hoppingivande	20.7 (9.3)
jovialisk	151.4 (0.1)	saker	0.0 (0.1)	inte bra	−115.7 (22.9)	artilleripjäs	20.6 (8.2)
förträffligt	151.4 (0.2)	släng	0.0 (0.1)	mossig	−120.6 (39.6)	matt	−8.7 (5.7)
festlig	135.4 (23.6)	överskatta	0.0 (0.1)	ointressant	−151.3 (0.2)	fatal	−8.7 (3.2)
lattjo	133.3 (21.2)	komma igång	0.0 (0.1)	utråkande	−151.4 (0.1)	nedgången	−8.7 (4.9)
lajban	132.6 (33.5)	ponera	0.0 (0.1)	trälig	−164.6 (18.6)	tungsinne	−8.9 (2.8)
roande	122.1 (22.0)	strosa	0.0 (0.2)	sorgset	−201.9 (0.2)	ålderstigen	−9.1 (6.2)
uppsluppen	113.7 (20.7)	förnimma	0.0 (0.1)	sorgen	−201.9 (0.1)	skruttig	−9.2 (4.2)
gladlynt	107.7 (22.8)	byta religion	0.0 (0.1)	neråt	−201.9 (0.1)	åldrig	−9.5 (4.0)
munter	102.4 (20.5)	drapera	0.0 (0.1)	boring	−216.9 (0.2)	flum	−9.7 (5.1)
gött	95.5 (25.7)	ytterst lite	0.0 (0.1)	ofördelaktig	−222.0 (0.2)	inkompetent	−9.8 (5.8)
...		...		deppad	−227.1 (0.1)	politik	−36.6 (7.5)

Table 1: Extract from lexicon. Average values for the most positve and negative words. We also present the words in the middle of the list, i.e. words deemed neutral, and some more examples. (Standard deviations for 10 repetitions of the method in Figure 1 within parentheses.)

1. For each word calculate v_+:
   - Repeat I number of times:
     - Walk randomly in the graph according to $\mathrm{prob}(s, d)$ for a maximum of m steps.
     - IF we hit a word in S_+ calculate the path length l using $\mathrm{len}(i, j)$, let $v_+ = v_+ + m/l$, and stop.
   - $v_+ = v_+/I$
2. For each word calculate v_- as in 1 with S_- instead.
3. For each word let $v = v_+ - v_-$

Figure 1: Random Walk. We use $I = 100$ and $m = 250$ and repeat all the above 10 times to calculate mean values and standard deviations.

The values have very different magnitudes. This may in part stem from that we use the synonymity level to define both transition probability and the length of the edges. The large standard deviations for some words are interesting. Perhaps they indicate that some words that should be connected are not.

If we only consider words with a polarity value bigger than their standard deviation we have 908 positive words and 441 negative words, this starting from only the very small lists of Section 3.

5. Conclusions and Future Work

From a small set of seed words we have constructed a first large, weighted polarity lexicon using the People's Dictionary of Synonyms. The lexicon consists of words from all word classes and different stylistic classes and could be a valuable resource for polarity classification in Swedish.

We will improve this work by considering larger and other sets of seed words. The seed words are not among the highest weighted words. One idea on how to address this is to include edges from each word to itself.

We intend to evaluate the lexicon by presenting positive, negative, and neutral words to human judges. The lexicon will become freely available.

6. References

Ahmed Hassan and Dragomir R. Radev. 2010. Identifying text polarity using random walks. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 395–403, Uppsala, Sweden, July. Association for Computational Linguistics.

V. Kann and M. Rosell. 2005. Free construction of a free Swedish dictionary of synonyms. In *Proc. 15th Nordic Conf. on Comp. Ling. – NODALIDA '05*.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135.

Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 777–785, Los Angeles, California, June. Association for Computational Linguistics.

LekBot
– A natural-language robot for children with communicative disabilities

Stina Ericsson
Dept. of Philosophy, Linguistics and Theory of Science
University of Gothenburg
&
Talkamatic
stina@talkamatic.se

1. Introduction

Children with communicative disabilities, for instance disabilities resulting from cerebral palsy or autism, have few opportunities to play independently and to interact on equal terms with children without communicative disabilities. One way in which this can be achieved is through a robot that is controlled by the child herself, on her own or together with other children. Internationally, there are a number of research projects that involve robots for children, including quite a few directed towards children with autism and other disabilities (Robins et al., 2008; Saldien et al., 2006; Kozima et al. 2007; Lee et al. 2008; Arent & Wnuk, 2007). However, none of these seem to involve communication through natural language in any form.

The LekBot project is a VINNOVA-funded collaboration between the University of Gothenburg, Talkamatic and DART. Acapela supports the project by providing their Acapela Multimedia TTS free of charge. LekBot started in March 2010 and runs until the end of 2011. The aim of LekBot is the development of a robot that uses current state-of-the-art technology to provide children, whether with or without communicative disabilities, with a toy that is easy and fun to use, and that involves natural language dialogue.

2. Communication and play

When playing with the LekBot robot, the child communicates by pressing buttons on a touch screen. The selected option is verbalised using a text-to-speech synthesiser, which acts as the child's voice. The robot communicates through its actions and linguistically also using a TTS. The precise characteristics, functionality and dialogical capabilities of the LekBot robot are to be determined during the course of the project. LekBot's predecessor, TRIK, was capable of drawing various objects on a sheet of paper on the floor (Ljunglöf et al., 2009), whereas LekBot will move around more freely, engaging with various objects in the room, and also include certain social and "physiological" capabilities, such as greet the user or indicate that it is tired or hungry.

At the time of writing, the current incarnation of LekBot can be told to go forward, go backwards or turn, and then carries out appropriate movements. When it goes forward and comes upon something that cannot be moved, such as a wall, it stops and variously exclaims "Oops!" ("Hoppsan!"), "Ouch!" ("Aj!") or "Wow!" ("Oj!"). If the user has not asked the robot to do anything during a specified amount of time – currently 20 seconds – the robot becomes bored, yawns, and starts to move around randomly for a while. This basic version of the robot thus allows the child to take some initiative, but can also take the initiative on its own.

3. System description

The heart of the LekBot system is the information-state based GoDiS dialogue manager (Larsson, 2002). The robot is built using Lego Mindstorms NXT, and currently includes a sensor for distance.

The child's communicative device is a communication board in the form of a touch screen that displays various symbols. Bliss symbols and Symbolstix are used for different children in the project. Acapelas's Swedish voices are used for the TTS, with different voices for the robot and for the user, that is, the child. Two sets of loudspeakers are used, one for the child's voice and one placed on the robot. The communication between computer and robot is via Bluetooth, rendering the use of an ASR superfluous. This means that "speech recognition" is always perfect, and that the natural language dialogue is there for the benefit of the child.

4. XP and user evaluations

LekBot development is done using Extreme Programming (Beck, 2005). XP practice involves programming in pairs, test-driven development and code refactoring, and of particular importance to the project, short iterations with frequent releases to the users.

During the first few months of the project, DART (a communication and computer resource centre for people with disabilities, and one of the three partners in the LekBot project) have acted in the interests of the users, specifying demands on the system and ranking proposed alternatives in the system's functionality.

The first release involving actual users is planned for October 2010. This will involve three pre-school children with cerebral palsy, and testing will take place at their respective daycare centres. The experiences of children and staff using LekBot will feed back into the development, and several such user evaluations during the project will help determine the robot's functionality and communicative behaviour. Each iteration will give priority to the development most beneficial to users.

5. Intonation and external events

Two areas of theoretical as well as practical interest in

the LekBot project, are intonation and external events. Both of these involve the extension of current dialogue models used by GoDiS. In the case of intonation, the TTS default pattern may need to be modified in order to render utterances as clear as possible, bearing in mind that erroneous or unclear intonation may pose a great challenge to children with cognitive disabilities. Models for improved intonation typically need to take dialogue context into account, as is explored for information-state models by Ericsson (2005).

External events concern the robot's movements through a changing environment. The system will need to handle external events coming from the robot, such as information that the robot is about to or has just hit an object. Such external events may lead to dialogue between the child and the robot, determining how the robot should handle the new situation.

6. Expected results

At the end of the project, a fully functional LekBot demonstrator will have been developed, which outwardly includes a communication board, a robot and a speech and symbol-based dialogue system. This demonstrator should be fun and user-friendly for children with communicative disabilities, encouraging children with disabilities to interact on their own with the robot, as well as together with a friend, and in both cases learning interactional skills through play. The demonstrator should also be easy to set up and control for day-care centre staff and other adults such as parents, and run in a robust way.

7. References

Arent, K. and Wnuk, M. (2007). Remarks on Behaviours Programming of the Interactive Therapeutic Robot Koala Based on Fuzzy Logic Techniques. In Proceedings of the 1st KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, 2007. Wroclaw, Poland. pp. 568 – 577.

Beck, K. (2005). Extreme Programming Explained: Embrace Change. 2nd ed. Boston, Addison-Wesley.

Ericsson, S. (2005). Information Enriched Constituents in Dialogue. PhD thesis. Dept. of linguistics, University of Gothenburg.

Kozima, H., Nakagawa, C. and Yasuda, Y. (2007) Children–robot interaction: a pilot study in autism therapy. *Progress in Brain Research* 164, pp. 385-400.

Larsson, S. (2002). Issue-based dialogue management. PhD thesis. Dept. of linguistics, University of Gothenburg.

Lee, C.H., Kim, K., Breazeal, C., Picard, R.W. (2008). Shybot: Friend-Stranger Interaction for Children Living with Autism. Paper presented at CHI, 5-10 April 2008. Florence, Italy.

Ljunglöf, P., Larsson, S., Mühlenbock, K. and Thunberg, G. (2009) TRIK: A Talking and Drawing Robot for Children with Communication Disabilities. Paper presented at NoDaLiDa, 14-16 May 2009. Odense, Denmark.

Robins, B., Dautenhahn, K., te Boekhorst, R. and Nehaniv, C.L. (2008). Behaviour delay and robot expressiveness in child-robot interactions: a user study on interaction kinesics. In Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction, 12-15 March, 2008. Amsterdam, The Netherlands. pp. 17-24.

Saldien J., Goris, K., Verrelst, B., Van Ham, R. and Lefeber, D. (2006). ANTY : The development of an intelligent huggable robot for hospitalized children. Paper presented at the 9th International Conference on Climbing and Walking Robots (CLAWAR), 12-14 September 2006. Brussels, Belgium.

Cocktail – a demonstration of massively multi-component audio environments for illustration and analysis

Jens Edlund, Joakim Gustafson, Jonas Beskow
KTH Speech, Music and Hearing
Lindstedtsvägen 24, SE-100 44 Stockholm
{edlund,jocke,beskow}@speech.kth.se

Abstract

We present MMAE – Massively Multi-component Audio Environments – a new concept in auditory presentation, and Cocktail – a demonstrator built on this technology. MMAE creates a dynamic audio environment by playing a large number of sound clips simultaneously at different locations in a virtual 3D space. The technique utilizes standard soundboards and is based in the Snack Sound Toolkit. The result is an efficient 3D audio environment that can be modified dynamically, in real time. Applications range from the creation of canned as well as online audio environments for games and entertainment to the browsing, analyzing and comparing of large quantities of audio data. We also demonstrate the Cocktail implementation of MMAE using several test cases as examples.

1. Introduction

In general, there are few methods for impressionistic inspection of large speech corpora around, although investigating such corpora is becoming increasingly important in many fields. There are few examples of techniques for overviewing large speech corpora, but an attempt worth mentioning is *tap2talk* (Campbell, 2003), which uses an entirely different approach than what we present here. One problem involved in immediate perceptualization of large amounts of speech, is that speech, as opposed to visual data, is transient and must be inspected in real time – we cannot easily listen to a static version, or a snapshot, of speech.

We are currently developing MMAE (Massively Multi-component Audio Environments), which offers a solution where large quantities of audio is inspected using simultaneous and dynamic playback of large numbers of distinct soundbites, creating a 3D soundscape reminiscent of a cocktail party. Here, we present Cocktail, the first version of a perceptualization engine based on this technique, to showcase some of its potential.

2. Background and motivation

MMAE has several uses. Here, we will talk about using it to mitigate the difficulties involved in perceptualizing large speech corpora, about using it as a tool for experimental analysis and comparison of speech, and about using it to create soundscapes that can be used in research as well as in entertainment.

2.1 Hearing the big picture

Getting a general feel for what a large speech corpus sounds like is useful both to researchers and service developers. Researchers are increasingly working on corpora that are so large the time required to listen through it sequentially could take more than a life-time. Moore (2003), for example, argues that given current development, error free ASR would require between 100 and 1000 years of acoustic training data. Although that is clearly not a feasible project, current ASR training routinely involves thousands of hours of acoustic data. And as automated speech services are becoming mainstream for customer care, the amount of speech data that must be analyzed to ensure quality of service is exploding. Many of these qualitative assessments require manual inspection, making exercises such as tracking quality of service and evaluating the effect of system changes an overwhelming task for service providers.

By playing a large number of sound clips from customers simultaneously, but at different locations in a virtual 3D space, researchers and developers can make judgments about the overall makeup of large databases quickly and get a feel of the mood of the simulated crowd. In addition, we are hoping to capitalize on the cocktail party effect, as described by Cherry (1953), who demonstrated that in settings such as a cocktail party, people are able to follow a conversation of their choice while ignoring others. This cocktail party effect can be utilized by listeners, who can focus on a particular speaker of the many simultaneous speakers for just as long as it takes to make a judgment, and then instantly skip to some other speaker. The effect is not created by the technology, but rather a case of technology taking advantage of how human speech perception works. In these ways, MMAE based perceptualization engines may provide near-instantaneous impressionistic inspection of large speech corpora.

2.2 Comparisons and analyses

In addition to the need to get a general impression of large speech corpora, there is a pressing need to be able to compare different speech corpora, or different subsets of the same corpus. The need arises for many reasons. We may want to compare

- sets of speech picked from different places in a dialogue – a serial comparison of data from different contexts that can potentially show problematic places.
- speech taken from the same context but different system versions - showing the effects of design choices;
- compare data collected at different times for regression and quality-of-service testing;
- compare perceived emotional state and user satisfaction to verify the soundness of new automatic methods to assess such subjective measures;

One way of achieving this would be to first listen to one subset, then to the next. As MMAE creates a 3D soundscape, we can aim for a more direct and efficient

method by offsetting one set of data to the left and the other to the right, and playing them simultaneously. Care must be taken to control for perceptual left/right ear differences, so each test should be conducted in both directions and experiment questionnaires should include questions about hearing left/right ear hearing impairments.

We have conducted pilot studies where listeners were asked to judge the ratio of males/females and speakers of two different dialects using cocktail vs. listening to short, sequential sound clips, we found the proposed method more accurate. The same tests indicate that the proposed cocktail method is considerably less stressful and cognitively more ergonomic to judges.

2.3 Soundscapes

The third and final motivation we will discuss here is further removed from pure speech technology. There are many cases where researchers and developers have a need to simulate acoustic environments involving huge numbers of sound sources: the noises of the jungle in a computer game, the boiling crowd of a football game, an audience clapping hands during a performance, or indeed the buzz of the participants of a cocktail party. Current auralization systems are generally not used to model such large quantities of sound sources. One reason is that it is computationally expensive to track and control great numbers of sound objects, another that it is easier to pre-record these soundscapes and place use them as backdrop for the more "important" foreground noises – a moving car or the main character speaking.

MMAE, however, can generate these soundscapes at relatively low computational cost. As a result, we can produce soundscapes that can be changed near-instantaneously, dynamically and online. The buzz of the cocktail party can increase and diminish, the crowd can grow silent in anticipation and burst into cheering at a goal, and the monkeys of a jungle can decide to become more or less noisy at any given time.

It is easy to see how these properties makes MMAE interesting from an entertainment industry point of view – games could potentially be provided with a more flexible sound environment at low cost. More static media, such as film and television, may also benefit from rapid and dynamic creation of complex soundscapes, for prototype purposes if nothing else. As we expect to see an increasing presence of spoken dialogue systems in games, there might be a certain overlap with speech technology applications here. Finally, MMAE simulations could provide very useful and realistic dynamically controllable crowd sounds for masking of speech in perception tests.

3. Technology

The Cocktail implementation of MMAE uses the Snack Sound library (http://www.speech.kth.se/snack/) as its backbone. The soundscapes consist of hundreds or even thousands of sounds played simultaneously and independently. This is made possible by relinquishing control over the individual sound objects on a number of levels, and even of their composition as a whole to some extent. For example, each sound is played fire-and-forget style – once playback has started, the sound is left to finish and disappear. Similarly, the composition of sounds, and the selection of what sound will be fired next, are not controlled in detail, nor do we keep track of it. Instead, sounds are selected from a repository using weighted random selection. The contents of the repository is configurable at initialization. The probabilities for a certain sound or class of sounds to be played is configurable during runtime, as is the probabilities of different positions in 3D space for each sound. Further, cocktail aims at keeping a certain number of sounds playing at any given time; this number is also configurable during runtime.

As an effect of the fire-and-forget policy, runtime changes of the soundscape – for example the composition of sounds, the number of sounds or their positioning – is not instantaneous. When a change is made to the configuration, it is applied only to sounds that are played after the change occurred. Sounds that are already playing are unaffected. For this reason, the latency of changes is dependant on the length of the sounds included in the repository: the shorter the sounds, the more responsive the changes. For this reason, applications such an applause machine are implemented using one single clap for each sound, making for very dynamic control. Similarily, babble simulations work better with short utterances or fragments than with longer utterances.

The most time consuming part of creating an experiment or a simulation, then, is to cut the sound into small enough pieces and label these, and then to describe the target composition of the soundscape. In the easiest case – say we only want to listen to a large amount of data quickly to get a first impression of it – this can be fully automatized. In more complex cases this work requires both thought and manual effort. We're currently developing tools to make the manual effort less taxing.

4. Summary

We have presented MMAE, a powerful and versatile technique for building dynamic, near-instantaneous 3D soundscapes by playing simultaneously and in chorus large numbers of short sounds. We have demonstrated a few of its applications in the Cocktail demo implementation, and discussed others. We believe that the technique can be a useful and valuable contribution, by itself or as a compliment to other techniques, in a large number of fields, including getting the big picture of speech or other acoustic databases; analyzing and comparing large sets of sounds; browsing and sorting speech and sound data; and creating dynamic simulations of environments with large numbers of sound sources such as cocktail parties, sports events, jungles, applause or traffic.

5. Reference

Campbell, N. (2003). tap2talk : an interactive interface for large speech corpora. In *Reports of the Meeting. the Acoustical Society of Japan* (pp. 223-224). Japan.

Cherry, E. C. (1953). Some experiments on the recognition of speech, with one and with two ears. *Journal of Acoustical Society of America, 25*(5), 975-979.

Moore, R. K. (2003). A comparison of the data requirements of automatic speech recognition systems and human listeners. In *Proceedings of Eurospeech 2003* (pp. 2582-2584). Geneva, Switzerland.

A Study of Rhetorical Strategies for Personalized Text Generation

Robert Krevers, Sture Hägglund
Department of Computer and Information Science, Santa Anna Research Faculty
Linköping University
robkr997@student.liu.se, sture.hagglund@santaanna.se

1. Problem background

The requirements for proper documentation of diagnostics and treatments have increased in the Swedish healthcare domain. As a result of this, many healthcare professionals need to devote a large amount of their time to documentation for the medical record, but they also need to produce adapted texts to be delivered to patients, employers, care givers, other health care personnel, social security agencies and insurance companies. Moreover, the law requires that the medical record itself should be written in a way that the patients can understand (SFS 2008:355 §13).

An automated text generation system or partially automated authoring system could hopefully aid in this endeavour, with the ability to handle vast amounts of information and quickly tailor texts according to specific parameters. Williams, Piwek & Power (2007) have created an example of such a system, which can turn the Electronic Medical Record (EHR) into a monologue or a (non-interactive) dialogue between two nurses in order to better explain the content of the EHR. While their system provide an interesting presentational structure and highlights the need of deeper explanation for patients, their system is limited to the goal of providing information. As a contrast, we believe that the system should be able to encompass emotions and willingness as well as knowledge goals. For this reason, we employ Rhetorical Structure Theory.

2. Rhetorical strategies in user-adapted texts

What's been done in the project so far is a study of how texts directed at healthcare professionals differed from those directed at patients, in order to investigate the basic presentational abilities that a system for mechanized generation of user-adapted texts from medical records would need. The study picked texts from FASS (*Farmaceutiska Specialiteter i Sverige*), a compilation of medication information that provides information of each medication both in a version directed to patients and in a version directed to healthcare professionals. The texts were analysed through **Rhetorical Structure Theory** (RST) in order to determine the kinds of rhetorical strategies used toward the different target groups.

RST is a well-known method for dividing a text into segments and mapping them into a hierarchical structure in relation to the other segments depending on what effect they cause in the reader of the text. The focus on the effect on the reader is one of the benefits of RST, which means that the analysts are not restricted to plain information content in comparison to user knowledge but can consider other user attributes as well, such as personality and emotions.

Ten text pairs with medical information from FASS were chosen at random, with each pair consisting of one text from Patients' FASS and one text from Physicians' FASS. The annotation program RSTTool (O'Donnel 2000) was used for segmentation of texts and relating the segments to each other in nucleus/satellite pairs or multinuclear relations of equal importance.

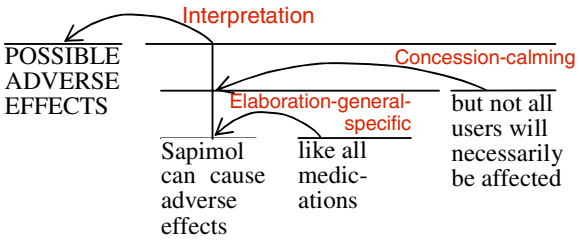


Figure 1: Excerpt from FASS annotated with RST relations in RSTTool (translated from Swedish by the authors).

The results of the study show that the texts directed to patients were more argumentative in its writing with direct instructions for the patient to follow. Typically, the texts for patients included examples of RST relations like *Motivation*, *Purpose*, *Enablement*, *Condition* and *Only-in-case*, all of which related to some action the reader should or should not perform. Texts directed to healthcare professionals on the other hand provide more comprehensive information to facilitate informed autonomous decision making, they were more extensive and provided a multitude of facts and details, often without providing explicit relations to the rest of the description (compact information rather than full text). The texts for physicians typically contained RST relations like *Evidence*, *Background*, *Reason* and *Summary*, which may provide and strengthen one or multiple opinions but the decision is left to the physician. Patients' FASS can be said to empower and instruct the reader in an argumentative fashion, while the Physicians' FASS guides the reader through rich and specific information that may contain hypotheses of varying certainty.

The RST relations and methods used were primarily based on the original work published by Mann & Thompson (1987) as well as a reference manual presented by Carlson & Marcu (2001). Out of all the relations suggested in those standard references, only a subset was found to be useful in the analysis of FASS medication texts. However, in the course of the analyses a number of additional relations were defined in order to better capture the intentions of some statements. These included some relations that are dealing with information focus, such as *Highlighting*, *Specification*, *Rhetorical-interception* and *Suggestion*, whereas others are more emotionally oriented in nature, such as *Alarm, Calming* and *Only-in-case*. The relations concerned primarily with information focus share similarities with existing relations, for instance *Highlighting* is similar to *Example*, but since it is only applicable when the example provided is especially important for the case at hand and thus provide additional detail. The emotional relations, on the other hand, appear to provide a different perspective on the 'effect of the reader' that is not clearly represented in the standard set of RST relations.

3. Text generation

As we proceed with the project, we will need to perform a more extensive study of rhetorical structures in relevant texts. However, the fact that the RST analyses of the two types of texts differed so much implies that RST can indeed be useful as a tool to capture and analyse rhetorical strategies to be used for adapting information to different recipients.

In the next step, our investigation of RST as a basis for managing rhetorical strategies for text generation will be extended to complete medical records. A prototype text generation system will be developed in order to allow user studies of synthesized texts as well. This will provide further insight about how rhetorical strategies can be used and what is needed to for them to be useful, for instance:

- What knowledge needs to be available for the system to know how to choose what rhetorical strategy to use?
- How is the choice performed? Under which conditions is a certain strategy chosen?
- What knowledge is necessary for the system to implement a rhetorical strategy?
- How do the rhetorical strategies interact?

Additionally, the differences observed in the FASS medication texts may be dependent on the different groups of readers having different goals, different pre-understandings of medical facts, different language capacities, different living situations or something else, by itself or in combination (see Cawsey, Grasso & Paris, 2007, for some excellent suggestions about what to include in a patient user model). This is another issue that requires further study to insure that the produced texts are optimal for the readers.

The project is part of the GenTex endeavour, where we are studying support for text generation from structured records in various application contexts. Studies include, in addition to rhetorical strategies, general methods for text generation, user modeling for tuning the generated texts to specific user needs and preferences, user studies of perceived text qualities, and design of supporting tools including methods for data collection, for instance structured interview techniques.

The produced text can either be a negotiated text where both parties fully understand the background and agree on the content, or two different versions expressing the same content but adapted to different types of users. In this case, there is typically one prime version of the text, which is formally valid, while a layman version may be produced to ensure a better understanding of the formal document.

Examples of situations where there is a need to present structured data records in the form of descriptive text in order to promote human communication and understanding are:

- Medical records to be read by people with different roles and background, such as physicians and patients.
- Records of structured interviews, where the interviewee need to understand and confirm the collected data.
- Requests and other statements submitted by filling in a form where a text presentation expresses the interpreted content of the completed form.
- Documentation of advisory consultations, such as investment advice from a financial advisor in a bank.

4. Reference

Carlson, L. and Marcu, D. (2001) Discourse Tagging Reference Manual. *ISI Tech Report* ISI-TR-545

Cawsey, A., Grasso, F. & Paris, C. (2007) Adaptive Information for Consumers of Healthcare. In *Lecture Notes in Computer Science*, Volume 4321

Mann, W. C. & Thompson, S. A. (1987) Rhetorical Structure Theory: A Theory of Text Organization. ISI: Information Sciences Institute, Los Angeles, CA, ISI/RS-87-190, 1-81. Reprinted from *The Structure of Discourse, L. Polanyi, ed.*

O'Donnell, M. (2000) RSTTool 2.4 - A Markup Tool for Rhetorical Structure Theory. In Proceedings of the 1st *International Conference on Natural Language Generation (INLG'2000)*.

SFS 2008:355 Svensk Författningssammling (2008) *Patientdatalag*, 2008-05-29.

Williams, S., Piwek, P. & Power, R. (2007) Generating monologue and dialogue to present personalised medical information to patients. In Proceedings of 11th *European Workshop on Natural Language Generation*

Detecting semantic innovation in dialogue

Staffan Larsson
University of Gothenburg
Sweden
sl@ling.gu.se

1. Introduction

Several mechanisms are available for semantic coordination (i.e., the process of interactively coordinating the meanings of linguistic expressions) in dialogue. These include corrective feedback, where one DP (Dialogue Participant) implicitly corrects the way an expression is used by another DP, as well as explicit definitions and negotiations of meanings. However, it also possible to coordinate silently, by DPs observing the language use of others and adapting to it. Adapting to semantically innovative language use requires, first of all, that the agent is able to detect semantic innovation. Towards this goal, this paper proposes a formal definition of semantic innovation.

We shall make use of type theory with records (TTR) as characterized in Cooper (2005) and elsewhere. The advantage of TTR is that it integrates logical techniques such as binding and the lambda-calculus into feature-structure like objects called record types.

2. Learning meaning from corrective feedback

We see corrective feedback as part of the process of negotiation of a language between two agents. Here is one of the examples of corrective feedback that we discuss in connection with our argument for this position in Larsson and Cooper (2009):

"Gloves" example (Clark, 2007):

- Naomi: mittens
- Father: **gloves**.
- Naomi: gloves.
- Father: when they have fingers in them they are called gloves and when the fingers are all put together they are called mittens.

In the Gloves example, after the father's utterance of "gloves", Naomi could use syntactic alignment to understand this term as a noun with the corresponding kind of compositional semantics:

$$\begin{bmatrix} \text{x} : \textit{Ind} \\ \text{c}_{\text{glove}} : \textit{glove'}(\text{x}) \end{bmatrix}$$

Provided that Naomi learns from the interaction that gloves are also a kind of clothing, Naomi's ontological semantics for "glove" after the first utterance by the father is the following type

$$\begin{bmatrix} \text{x} & : & \textit{Ind} \\ \text{c}_{\text{glove}} & : & \text{glove}'(\text{x}) \\ \text{c}_{\text{physobj}} & : & \text{physobj(x)} \\ \text{c}_{\text{clothing}} & : & \text{clothing(x)} \end{bmatrix}$$

3. Perceptual type

We here add a further aspect of meaning, namely *perceptual type* (or perceptual meaning). For our current purposes, we will represent perceptual meaning as a record type specifying and individual and one or more propositions indicating that the individual is of a certain perceptual type, i.e., that it has certain physically observable characteristics.

The word "glove'', for example, may be associated with a certain shape:

$$\begin{bmatrix} \text{x} : \text{Ind} \\ \text{c}_{\text{glove}-\text{shape}} : \text{glove-shape(x)} \end{bmatrix}$$

4. Contextual interpretation

To represent individual dialogue participants' takes on contexts[1], we will use record types with manifest fields. This allows our context to be underspecified, reflecting the fact that an agent may not have a complete representation of the environment.

For our current purposes, we assume that our DPs are able to establish a shared focus of attention, and we will designate the label "focobj" for the object or objects taken by a DP to be in shared focus.

We will first show how "normal" contextual interpretation, in the absence of innovations, is assumed to work. We will assume that parts of the meaning of an utterance are *foregrounded*, whereas other parts are *backgrounded*. Background meaning (BG) represents constraints on the context, whereas foreground material (FG) is the information to be added to the context by the utterance in question. We can represent this either as a record or as a function:

$$\begin{bmatrix} \text{BG} = \ldots \\ \text{FG} = \ldots \end{bmatrix}$$

$$\lambda t \sqsubseteq \text{BG} \,.\, t \dot{\wedge} (\text{BG} \dot{\wedge} \text{FG})$$

The functional version takes as argument a record type t, representing the current context, which is a subtype[2] of the

[1] Occasionally and somewhat sloppily referred to as "contexts" below.

[2] Formally, $T_1 \sqsubseteq T_2$ means that T_1 is a subtype of T_2. Two examples will suffice as explanation of this notion:

background meaning of the uttered expression (typically a context containing manifest fields representing objects in the environment and propositions about these objects). The function returns a record type corresponding to the union of t and the union of the background and foreground meanings.

5. Formalising innovation

This section provides a TTR analysis of detection of innovative language use. We will focus on the case where a known expression is used with a (subjectively) innovative meaning. The underlying intuition is that the meaning of an expression should say something about the kind of context in which it can be (non-innovatively) used. But how, exactly? Here is our proposal.

An expression e is *innovative* in context c if there is a mismatch between e and c in either of the following ways:

1. *Background inconsistency*: Some information presupposed by the expression contradicts some information in the context; formally $[e].\text{BG} \dot{\wedge} c \approx \bot$
2. *Foreground inconsistency*: Some content conveyed by the utterance of the expression contradicts something in the context; formally $[e](\text{c}) \approx \bot$

This definition follows naturally from how contextual interpretation works. Recall that meaning can be seen as a function from context to content, where background meaning serves as a constraint in the context. The definition of innovation checks that it will be possible to apply the meaning-function to the context, by checking that the context is consistent with the constraints imposed by the backgrounded meaning, and that the resulting contextual interpretation will be consistent with the context.

As an example of detection of innovation we will use a modified version of the "gloves" example, where Naomi simply observes an utterance by Father:

Modified "Gloves" example:

- (Naomi is putting on her new gloves)
- Father: Those are nice gloves!

Here, we wish to illustrate what happens when a previously known word is encountered with a different meaning. We therefore assume, for the sake of argument, that Naomi initially has a concept of gloves. We will assume that Naomi takes "gloves" as having a perceptual type distinct for that of "mittens". However, again for the sake of argument, we assume that she is mistaken as to the nature of this difference; for example, she may disregard the difference in shape and instead think that mittens and gloves have different textures (e.g. that gloves are shiny whereas mittens are woolly).

$$\begin{bmatrix} \text{ref} : \text{Ind} \\ \text{c} : \text{glove(ref)} \end{bmatrix} \sqsubseteq \begin{bmatrix} \text{ref} : \text{Ind} \end{bmatrix}$$

$$\begin{bmatrix} \text{ref=obj123} : \text{Ind} \end{bmatrix} \sqsubseteq \begin{bmatrix} \text{ref} : \text{Ind} \end{bmatrix}$$

$\bigwedge[\text{glove}]_{Naomi} =$

$$\begin{bmatrix} \text{x} : \textit{Ind} \\ \text{c}_{\text{glove}} : \text{glove}'(\text{x}) \\ \text{c}_{\text{physobj}} : \text{physobj(x)} \\ \text{c}_{\text{clothing}} : \text{clothing(x)} \\ \text{c}_{\text{shiny-texture}} : \text{shiny-texture(x)} \\ \text{c}_{\text{handclothing-shape}} : \text{handclothing-shape(x)} \end{bmatrix}$$

That is, Naomi thinks that mittens and gloves both have a common shape, but that they differ in texture. This means that the meaning of Father's utterance will be

$[\text{Those are nice gloves}]^{Naomi} =$

$$\begin{bmatrix} \text{BG} = & \begin{bmatrix} \text{focobj} : \textit{Ind} \\ \text{c}_{\text{glove}} : \text{glove}'(\text{focobj}) \\ \text{c}_{\text{physobj}} : \text{physobj(focobj)} \\ \text{c}_{\text{clothing}} : \text{clothing(focobj)} \\ \text{c}_{\text{shiny-texture}} : \text{shiny-texture(focobj)} \\ \text{c}_{\text{handclothing-shape}} : \text{handclothing-shape(focobj)} \end{bmatrix} \\ \text{FG} = & \begin{bmatrix} \text{c}_{\text{nice}} : \text{nice}'(\text{FG.focobj}) \end{bmatrix} \end{bmatrix}$$

When encountering Father's utterance, we take it that the relevant take on the context for evaluating and understanding the utterance is something like

$c^{Naomi} =$

$$\begin{bmatrix} \text{focobj=a} : \text{Ind} \\ \text{c}_{\text{physobj}} : \text{physobj(focobj)} \\ \text{c}_{\text{clothing}} : \text{clothing(focobj)} \\ \text{c}_{\text{woolly-texture}} : \text{woolly-texture(focobj)} \\ \text{c}_{\text{handclothing-shape}} : \text{handclothing-shape(focobj)} \\ \text{c}_{\text{not-shiny-texture}} : \text{not(shiny-texture(focobj))} \end{bmatrix}$$

The $\text{c}_{\text{not-shiny-texture}}$ field can either result from consulting the environment by checking whether a shiny texture cannot be detected on focobj, or by inference from the proposition in $\text{c}_{\text{woolly-texture}}$.

Now, according to our definition of innovation, Naomi will detect a background inconsistency in that [Those are nice gloves].$\text{BG} \dot{\wedge} c_{Naomi} \approx \bot$. The inconsistency of course stems from the presence of a proposition (shiny-texture(focobj)) and its negation in the combined record. Contextual interpretation will thus fail, since the meaning-function cannot be applied to the context.

Acknowledgments

This research was supported by The Swedish Bank Tercentenary Foundation Project P2007/0717, Semantic Coordination in Dialogue.

6. References

E. V. Clark. 2007. Young children's uptake of new words in conversation. *Language in Society*, 36:157–82.

Robin Cooper. 2005. Austinian truth, attitudes and type theory. *Research on Language and Computation*, 3:333–362.

Staffan Larsson and Robin Cooper. 2009. Towards a formal view of corrective feedback. In Afra Alishahi, Thierry Poibeau, and Aline Villavicencio, editors, *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition, EACL*, pages 1–9.

Using Stylistic Differences in Information Retrieval

Karin Friberg Heppin

Språkbanken, Department of Swedish Language
University of Gothenburg
karin.friberg@svenska.gu.se

Abstract

The MedEval test collection is a newly constructed Swedish medical test collection, unique in its kind providing the possibility to choose user group: doctors or patients. A test collection such as MedEval makes it possible to study how to construct queries in order to retrieve documents intended for one group or the other.

1. MedEval

The MedEval test collection is built on documents from the MedLex medical corpus (Kokkinakis, 2004). MedLex consists of scientific articles from medical journals, teaching material, guidelines, patient FAQs, health care information, etc. The set of documents used in MedEval is a snapshot of MedLex in October 2007, approximately 42 200 documents or 13 million tokens.

MedEval is a Swedish medical test collection where assessments have been made, not only for topical relevance, but also for target reader group: Doctors or Patients. The user of the test collection can choose if she wishes to search in the Doctors or the Patients scenario where the topical relevance assessments have been adjusted, or in a scenario which regards only topical relevance. This enables the user to compare the effectiveness of searches retrieving documents aimed at one group or the other. MedEval is the first Swedish medical test collection.

The doctoral thesis Friberg Heppin (2010) describes the construction of the MedEval test collection. It also describes pilot studies demonstrating how such a collection may be used. The MedEval test collection has:

- documents assessed on a four-graded (0-3) scale of relevance allowing a fine-grained study of retrieval effectiveness.
- documents assessed for target reader group allowing studies of document retrieval based on topic relevance as well as on intended audience.
- documents marked for target reader group allowing studies of differences in the language registers.
- the potential of being a valuable resource in teaching in language technology, information retrieval and linguistics.

2. Target Groups

For a classification of documents according to intended readers to be useful, there must be measureable differences between the classes. Table 1 shows a number of type/token frequencies in subsets of the collection. These are described below. In each subset duplicates were removed if a document had been assessed for more than one topic. Full form types are the original terms of the documents and lemma types are the same terms after lemmatization.

Entire collection All documents in MedEval.

Assessed documents All documents that have been assessed for any topic.

Doctors All documents that for at least one topic have been assessed to have target group Doctors.

Patients All documents that for at least one topic have been assessed to have target group Patients.

Common files All documents that for at least one topic have been assessed to have target group Doctors and for another to have target group Patients.

Doctors relevant All documents that for at least one topic have been assessed to have at least relevance grade 1 and to have target group Doctors.

Patients relevant All documents that for at least one topic have been assessed to have at least relevance grade 1 and to have target group Patients.

Table 1 shows that the patients' documents had only 57% of the doctors' number of tokens per document. Even though there were over 1 000 more patient than doctor documents, there were over 50 000 more lemma types in the doctor documents and almost 30 000 more lemma compound types. The average word length for doctors was 6.29 compared to 5.73 for patients. The ratio of compound tokens was also higher for doctors, 0.128 compared to 0.098.

There is a clear difference in the type-token ratio of the subsets of MedEval. In Patients assessed the ratio is 33.2 compared to 25.6 in Doctors assessed, even though there are 800 000 more tokens in the Doctors set. Bearing in mind that type-token figures are dependent on the size of the collection, the result is even more noteworthy.

3. User Groups

The MedEval test collection allows the user to state user group: *None*, *Doctors* or *Patients*, directing her to one of three scenarios. The None scenario contains the topical relevance grades as made by the assessors. The Doctors scenario contains the same asessments but with the relevance

	Entire collection	Assessed documents	Doctors assessed	Patients assessed	Common files	Doctors relevant	Patients relevant
Number of documents	42 250	7 044	3 272	4 334	562	1 233	1 654
Tokens	12 991 157	5 034 323	3 232 772	2 431 160	629 609	1 361 700	988 236
Tokens/document	307	715	988	561	1 120	1 104	596
Average word length	5.75	6.04	6.29	5.73	6.16	6.33	5.63
Full form types	334 559	181 354	154 901	92 803	50 961	87 814	43 825
Lemma types	267 892	146 631	126 217	73 121	40 857	71 974	34 263
Lemma type token ratio	48.5	34.3	25.6	33.2	15.4	18.9	28.8
Compound tokens	1 273 874	573 625	412 475	237 267	76 117	179 580	92 420
Full form compound types	187 904	99 614	83 846	47 387	24 083	45 257	20 157
Lemma compound types	144 159	78 508	66 907	37 151	19 685	36 867	16 006
Ratio of compounds	0.098	0.114	0.128	0.098	0.120	0.132	0.094

Table 1: Type and token frequencies of the terms in different subsets of the MedEval test collection.

of the documents marked for Patients target group downgraded by one. In the same way the Patients scenario has documents marked for Doctors downgraded by one.

To demonstrate the effectiveness of search terms from the different registers, two synonyms for 'anemia', *anemi*, and *blodbrist* were run as search keys in the Doctors and Patients scenarios for one topic. *anemi* is a neoclassical term belonging to the expert language and *blodbrist* is the corresponding lay term. The results are shown in table 2.

In the Doctors scenario the difference between the results of the two search keys was striking: full recall for the neoclassical term quite early in the ranked list of documents and no recall at all for the lay term. In the Patients scenario, the neoclassical term did not perform quite as well as it did for doctors, and the lay term did not perform as bad as it did for doctors. Note that the resulting ranked lists of documents is the same for both scenarios for the same search key. It is the relevance grades of the retrieved documents that differ.

Scenario	Recall	anemi	blodbrist
Doctors	@10	50% (4/8)	0% (0/8)
	@20	100% (8/8)	0% (0/8)
	@100	100% (8/8)	0% (0/8)
Patients	@10	22% (4/18)	33% (6/18)
	@20	39% (7/18)	39% (7/18)
	@100	66% (12/18)	56% (10/18)

Table 2: Running the synonyms *anemi* and *blodbrist* as search keys in the Doctors scenario gave full recall early in the ranking for the neoclassical term, but no recall for the lay term. In the Patients scenario the difference in effectiveness was not as striking.

One plausible reason for the different results is that experts do not use lay terms. These are often imprecise and can even be misleading. An example is *blodbrist*. Even though the literal meaning is 'blood deficiency' the term does not refer to a deficiency of blood, rather a deficiency of red blood cells or of hemoglobin. In contrast, lay texts often contain both lay and expert terms. An expert term may be used, and a lay term added as an explanation, or a lay term may be used and an expert term presented as additional information. Both examples are shown in figure 1. It is interesting that the patient documents often contain medical terms from both registers bearing in mind that they contain fewer types than the doctor documents.

> `B12 är ett vitamin som är nödvändigt för bildningen av röda blodkroppar, brist kan då ge det vi kallar perniciös anemi (anemi betyder just blodbrist).`
> B12 is a vitamin that is necessary for the production of red blood cells, deficiency can cause what we call pernicious anemia (anemia means precisely blood deficiency).
>
> `...t.ex. fel på sköldkörteln, diabetes eller en speciell form av blodbrist , s.k. perniciös anemi.`
> ...e.g. failure of the thyroid gland, diabetes or a special form of blood deficiency, known as pernicious anemia.

Figure 1: Two examples of synonyms from different registers used in one sentence. In the first example the lay term is used as an explanation, and in the second the expert term is supplied as additional information.

4. Final Words

The main novelty of MedEval is the marking of target groups, Doctors and Patients, together with with the possibility to choose user group. This opens up new areas of research in Swedish information retrieval such as how one can retrieve documents suited for different groups of users.

The Department of Swedish Language at the University of Gothenburg is in the process of making the MedEval test collection available to academic researchers.

5. References

Karin Friberg Heppin. 2010. *Resolving Power of Search Keys in MedEval a Swedish Medical Test Collection with User Groups: Doctors and Patients*. Ph.D. thesis, University of Gothenburg.

Dimitrios Kokkinakis. 2004. MEDLEX: Technical report. Technical report, Department of Swedish, University of Gothenburg, ⟨http://demo.spraakdata.gu.se/svedk/pbl/MEDLEX_work2004.pdf⟩.

Is *data scrubbing* capable of protecting the confidentiality and integrity of sensitive data?

Dimitrios Kokkinakis
Centre for Language Technology (CLT) & Språkbanken, Department of Swedish
University of Gothenburg, Box 200, 405 30 Göteborg
dimitrios.kokkinakis@svenska.gu.se

Abstract

The release of individual data for research, public health planning, health care statistics, monitoring of diagnostic tests, automated data collection for health care registries and tracking disease outbreaks are some of the areas in which the protection of Personal Health Information (PHI) has become an important concern. The purpose of this study is to adapt and apply synergetic methods to document de-identification, particularly in the clinical setting. The main challenge is to retain important concepts and PHI in the documents in a standardized and neutral manner as means of encryption without violating the integrity of the PHI and without sacrificing the quality and intended meaning of the authors.

1. Introduction

De-identified data can be used as a source of information and knowledge to broad spectrum of services related to the demands for dissemination of confidential information about individuals. In the clinical setting for instance, hospitals continuously produce, manage and store vast amounts of patient-related data. Due to confidentiality requirements these data – mostly in textual form – remain inaccessible for research and knowledge mining (Pestian *et al.*, 2006). The need of disclosure of personal health information (PHI) in Electronic Health Records (EHR) for secondary purposes (Safran *et al.,* 2007) is expected to increase dramatically the coming years.

2. Background

Access of clinical text for research purposes which can ensure protection of PHI, can be either granted by the patients themselves, by obtaining permission from institutional review boards, or by data use agreements. In any case de-identification of various explicit identifiers (such as names of relatives) is often required. De-identification is defined as the process of recognizing and deliberately changing, replacing or concealing the names and/or other identifying information of relevance about entities, generally called PHI, from clinical or other sensitive to disclosure data. Data scrubbing is another term used for the same purpose (*cf*. Sweeney, 1996) sometimes with a bit lower understandability ambitions (Berman, 2003). In our context we do not make a differentiation between these two terms which we consider as synonymous.

Many de-identification techniques are described in the literature. One of the earliest systems, the "Scrub" system (Sweeney, 1996), is based on a set of detection algorithms utilizing word lists and templates that each detect a small number of name types in pediatric records. The *k-anonymisation* approach Sweeney (2002), de-associates attributes from corresponding identifiers (e.g. date of birth), each value of an attribute is suppressed or generalized. A recent review on de-identification in the clinical domain is given in Meystre *et al.* (2008); while one of the first publicly available de-identification software and relevant test data are described in Neamatullah *et al.* (2008). Uzuner *et al.* (2006) present details of systems participated in a shared task of automatic de-identification of medical summaries. For Swedish, which is our application language, it is worth mentioning the works by Kokkinakis & Thurin (2007) and Velupillai *et al.* (2009).

2.1 HIPAA and PHI

In different parts of the world confidentiality is regulated and protected by various mechanisms such as the US Health Insurance Portability and Accountability Act (HIPAA, 2003). Such policies state that for a text to be rendered as safely de-identified, information such as e.g. names, dates, etc. must be removed. In many cases such information has been modified to suit different needs. HIPAA defines 18 different data elements that should be replaced from any type of sensitive data in order for them to be considered de-identified. Researchers such as Hrynaszkiewicz *et al*. (2010) discuss a 28 item list of patient identifiers some of which are complementary to the previous (e.g. biometric data), while Velupillai *et al.* (2009) discuss that e.g. ethnicity might also be another identifier that can reveal crucial to re-identification identifiable information. To these identifier lists, rare disease names, certain forms of ethnic clothing or exceptional personal qualities may also be added.

3. Materials and Methods

Ethical issues might be a barrier to directly accessing patient data, without approval from ethical committees. Therefore we explore similar texts given to medical students in the form of written examination papers. These texts mirror the reality the students are suppose to meet as they start their professional career; these reports are considered equivalent to real reports in terms of the PHI

that can be identified. For the study we have assembled a corpus of 52 EHR-like reports (150000 tokens) from medical faculties across Sweden.

The following methodology has been applied to the corpus, originally inspired by the work of Berman (2003): (i) terminology recognition; (ii) extended named entity recognition; (iii) bias introduction (optional) and (iv) data scrubbing. Data scrubbing: from a large newspaper text corpus we extracted two lists of the 5000 and 10000 most frequent tokens. During processing if a token in a test text is among the 5000 (or 10000) most frequent ones it is kept in tact. Also, punctuation markers, numbers of length <3 and tokens consisting of 1 character are also kept intact. All other tokens that are not part of the previous steps are scrubbed according to their length and orthographic characteristics. Characters are changed to an asterisk '*', and numbers to 'N', both respecting the tokens' length.

4. Evaluation Issues and Conclusions

The main goal of this work is to retain important concepts and PHI in the documents in a standardized and neutral manner without violating the integrity of the PHI and without sacrificing the quality and intended meaning of the authors. In a small scale evaluation conducted (by the author) the results showed that roughly all sensitive PHI in these texts have been either replaced by neutral labels by the NER process or scrubbed rendering the textual data harmless (the evaluation material can be found here: <http://demo.spraakdata.gu.se/svedk/pbl/scrubbCorpusText.txt>).
The results seem adequate for fulfilling the first goal and we have implemented an interface that anyone can use for testing the validity of these results.

With respect to the second goal, information preservation, intended meaning, the scrubbing approach is sensitive to the amount of the general language that can be retained in the original texts. In many cases there is a risk that the meaning of a sentence is changed or lost. The 10000 word limit seems an acceptable threshold for this purpose (the top-5000 tokens seems too limited) however a higher threshold might be more suitable for information preservation. Unsafe terms in the threshold lists, e.g. *rape* can be excluded during the scrubbing process (not implemented yet). The methodology outlined revealed a number of limitations. Some domain-specific acronyms have been scrubbed which implies that some valuable information might be lost, data cleansing (acronyms expansion) might be necessary in order not to lose valuable information. There were also medical terms that we could identify as *scrubbed*, this depends on either the limitations of the standardised taxonomies applied with respect to their coverage or because of misspellings or ad hoc variant term forms. Different methods for de-identification of sensitive data must be sought since it is a known fact that manually removing PHI is a time consuming and costly enterprise. The difficulty of the task is illustrated by Dorr *et al.* (2006), where they point out that even simple PHI is difficult to automatically identify with the exactitude required by HIPAA. Also, a major problem that has been recently recognized is the lack of metrics that can quantify the risk of re-identification and information preservation using different de-identification techniques Hirschman & Aberdeen (2010).

The first goal of our work is easier to evaluate the second harder. A demonstration interface has been implemented that illustrates the functionality of the scrubbing process; the user has the possibility to test the text understandability by choosing appropriate values that can be used for refining the evaluation. Therefore we let human subjects read the results and grade in a scale how well they understand the resulted scrubbed text. In the future we intend to integrate and combine more standardized resources in order to achieve a higher lever of understanding and experiment with other thresholds.

5. Reference

Berman J.J. (2003). Concept-Match Medical Data Scrubbing. *Arch Pathol Lab Med*, 127: 680-686.

Dorr DA. *et al.* (2006). Assessing the difficulty and time cost of de-identification in clinical narratives. *Methods Inf Med*. 45(3):246-52.

Hirschman L. & Aberdeen J. (2010). Measuring Risk and Information Preservation: Toward New Metrics for De-identification of Clinical Texts. Text & Data Mining of Health Documents, NAACL workshop. LA, CA.

Hrynaszkiewicz I., Norton ML., Vickers AJ. & Altman, DG. (2010). Preparing Raw Clinical Data for Publication: guidance for journal editors, authors, and peer reviewers. *Trials* 11:9.

Kokkinakis D. & Thurin A. (2007). Anonymisation of Swedish Clinical Data. The 11th Conf. on AI in Medicine (AIME). Pp. 237-241. Netherlands.

Meystre SM., Savova GK., Kipper-Schuler KC. & Hurdle JF. (2008). Extracting Information from Textual Documents in the EHR: a review of recent research. *Yearb Med Inf*. 128-44.

Neamatullah I., *et al.* (2008). Automated De-Identification of Free-Text Medical Records. *BMC Medical Informatics and Decision Making*, 8:32.

Pestian JP., Itert L., Andersen C. & Duch W. (2006). Preparing Clinical Text for Use in Biomedical Research. *J of Database Management*, 17(2), 1-11.

Safran C., et al. (2007). Toward a National Framework for the Secondary Use of Health Data: An Am Med. Inf Assoc. White Paper. JAMIA 14:1-9

Sweeney L. (1996). Replacing Personally-Identifying Information in Medical Records, the Scrub System. *J. of the Am Med Informatics Assoc.* Pp. 333-337.

Sweeney L. (2002). k-anonymity: a Model for Protecting Privacy. *J. Uncertain. Fuzziness Knowl.-Based Syst.* 10(5): 557-570.

Uzuner O., Kohane I. & Szolovits P. (2006). Challenges in NLP for Clinical Data Workshop. <http://www.i2b2.org/NLP/Schedule-final.pdf>

Velupillai S., Dalianis H., Hassel M. & Nilsson G. (2009). Developing a Standard for de-identifying Electronic Patient Records Written in Swedish: Precision, recall and F-measure in a Manual and Comput. Annot. Trial. *J. of Med. Inf.* vol. 78:12, pp. e19-e26.

Creating a reusable English – Afrikaans parallel corpora for bilingual dictionary construction

Aldin Draghoender, Mattias Kanhov
Department of Computer and Systems Science, (DSV)
Stockholm University
Forum 100, 164 40 Kista, Sweden
aldr-dra@dsv.su.se, kanhov@dsv.su.se

Abstract

This paper investigates the possibilities in creating a bilingual English – Afrikaans dictionary by building a parallel corpus and using the Uplug tool to process it. The resulting parallel corpus with approximately 400,000 words per language was created partly from texts collected from the South African government and partly from the OPUS corpus. The recall and accuracy of the bilingual dictionary was evaluated based on the statistical data collected. Samples of translations were generated, compiled as questionnaires and then assessed by English – Afrikaans speaking respondents. The results yielded an accuracy of 87.2 percent and a recall of 67.3 percent for the processed dictionary. Our English – Afrikaans parallel corpora can be found at the following address: http://www.let.rug.nl/tiedeman/OPUS/

1. Introduction

Whether it is for business intelligence, shopping or for communicating in social websites such as Facebook, the Internet has become the largest source of information thus creating a platform for multilingual information retrieval. South Africa is a country with eleven official languages where most of the population only speaks a small percentage of all the languages and could therefore benefit from multilingual information retrieval. For this reason the need of a multilingual dictionary is of great importance.

In this paper we present our work where we created a parallel corpus, ran it through the Uplug tool, generated a dictionary and then finally processed and evaluated it.
Previous research using Uplug for word alignment of parallel corpora was performed by for example Dalianis et al (2009) with 71 percent average frequency and an average recall of 93 percent on Swedish - English. There was also no confirmation that POS-tags improve word alignment. Charitakis (2007) had a Greek-English parallel corpus which comprised of about 200 000 words per language. The conclusion based on their quality was that 51 percent (f>3) of the translations were correct while with higher frequency (f>11) 67 percent was achieved.

2. Creating a reusable corpus

Because of the lack of parallel corpora, we decided to create our own corpus by mining multiple English – Afrikaans bilingual texts from the Internet. However, during the corpus creation process we received a portion of the OPUS corpus by Tiedemann and Nygaard (2004).

This meant that our final corpus would be partly from the OPUS corpus and partly from a parallel corpus that we created by sourcing publications from the South African government website (South African Government Information, 2010). These publications were converted from PDF format to plain text and then manually aligned at paragraph level. Only small modifications were needed after that as the texts already were aligned at sentence level for the most part. The final corpus contained 421,587 Afrikaans words and 397,757 English words respectively and covering three domains: Law, public speeches and technical documentation. Around 200,000 words (roughly 50%) per language originated from the OPUS corpus.

3. Uplug and word alignment

The Uplug system is an application with the purpose of providing a modular platform for the integration of text processing tools (Uplug, 2010). The reason why Uplug was the system of choice is because it has been used in many similar projects and it is fairly easy to get acquainted with. The resulting dictionary contained a total of 87,388 lines of word pairs (translations) with one pair per line after a total runtime of 9 hours 22 minutes and 54 seconds. The dictionary however contained many duplicate words and punctuation mark translations, so it needed to be cleaned. The cleaning was done manually because the errors in the dictionary were often unique, making automated cleaning difficult to configure. The translations with frequency of 2 or less were seen as unreliable and therefore removed from the dictionary.

After removing these duplicates and words with a frequency of 2 or less, we finally got a "cleaned" dictionary with 6,450 word pairs which was a 91 percent decrease from the original size.

4. Evaluation

Finally to evaluate the original- and cleaned dictionary, three different sample texts in English were used along with three different types of measuring techniques. The sample texts were chosen as to cover several domains in order to get reliable results. The following measuring techniques were used:

English words found – to measure the amount of words from the sample texts which were present in the dictionary.

Accuracy – the amount of words found in the sample texts that were present in the dictionary and were correctly translated. The words not found in the dictionary would be ignored.

Recall – the amount of correctly translated words that were found in the sample texts. The words not found would be considered as incorrect translations.

Dictionary	English words found in dictionary	Accuracy	Recall
Original	85.48%	79.11%	71.71%
Cleaned	75.27%	87.16%	67.31%

Table 1. The summarized results.

We compiled a questionnaire from the English words found and their translations that were evaluated by English/Afrikaans speaking respondents as well as Google Translate. The respondents evaluated the word pairs by deeming them either Correct, Partly correct or Wrong.

These results were then used to calculate *accuracy* and *recall*. Google Translate was used because of the small number of evaluating people. The English translation of the word pairs was entered into the translator, if the translation corresponded to the Afrikaans word in the word pair they were considered correct. If the translator produced a different word, that word was then entered into Google Translate. If the English word produced corresponded to the English word in the word pair, it was considered correct or partly correct depending on the accuracy.

5. Results

The average values for the evaluations done of the original and cleaned dictionary are seen in Table 1.

Evaluator	Correct	Partly correct	Wrong
Google translate	85.26%	6.17%	8.57%
Person A	87.35%	8.04%	4.61%
Person B	91.04%	5.91%	3.06%
Person C	91.37%	4.86%	3.77%
Person D	80.77%	5.32%	13.91%
Average	**87.15%**	**6.06%**	**6.78%**

Table 2. Accuracy evaluations for the cleaned dictionary.

The decrease of English words found is understandable as the majority of the translations in the dictionary are low frequency and therefore removed during the cleaning process.

The accuracy for the cleaned dictionary had an average improvement of around 8 percentage points compared to the original dictionary, showing the importance of manual dictionary cleaning.

6. Conclusions and future work

When creating a parallel corpus, we found that many errors can occur when PDF documents are converted to plain text, therefore it is important that the whole text is thoroughly reviewed to identify errors. The texts must also manually be paragraph aligned (and preferably also sentence aligned) to get a good result but it demands a lot of time as most corpora are composed of several thousand sentences or more.

Uplug was a very effective tool when processing the corpus. Except for some duplicate- and double translations as well as an error with wrong character encoding, the whole process worked very well.

The results showed a clear connection between how many English words found from the sample texts, recall and accuracy when comparing the original dictionary with the cleaned one. The size of the dictionary was reduced to 9 percent of its original size after cleaning it, the amount of English words found was reduced to 75.5 percent from the original 85.5 percent while the accuracy increased from 79.1 percent to 87.2 percent, showing that a huge number of the translations with frequency of 2 or less were faulty and unnecessary.

The fact that Afrikaans is closely related to English and in addition to a large corpus, we got a relatively high overall accuracy compared to similar research. We also found that manually processing and cleaning the dictionary is an important step to ensure high accuracy.

For future work, a good idea may be to use a lemmatizer to get the base form of the word which could lead to better results. As we did not find an Afrikaans lemmatizer, one idea could be to use a Dutch lemmatizer since the languages share the same language structure.

For further reading see Draghoender & Kanhov (2010).

Acknowledgement

We would like to thank our supervisor Hercules Dalianis for his support and our respondents who answered the translation questionnaires.

References

Charitakis, K. (2007). Using parallel corpora to create Greek -English dictionary with Uplug. In Proc. of Nodalida, 2007, 16th Nordic Conference of Comp. Ling., 25-26 May 2007. Tartu, Estonia.

Dalianis, H., Rimka, M. and Kann, V. (2009). Using Uplug and SiteSeeker to construct a cross language search engine for Scandinavian languages. In Proc. of Nodalida, 17th Nordic Conference on Comp. Ling, May 15-16 2009. Odense, Denmark.

Draghoender, A. and Kanhov, M. 2010. Creating a reusable English - Afrikaans parallel corpora for bilingual dictionary construction, B.Sc thesis. Department of Computer and Systems Sciences, (DSV), Stockholm University.

South African Government Information. (2010). [Online] Available at http://www.info.gov.za/ [Accessed 17 March 2010].

Tiedemann, J. and Nygaard, L. (2004).The OPUS corpus - parallel and free. In Proc. of the Fourth International Conference on Language Resources and Evaluation, (LREC), May 26-28, 2004. Lisbon, Portugal.

Uplug, (2010). The Uplug homepage. [Online] Available at: http://www.let.rug.nl/~tiedeman/Uplug/ [Accessed 20 January 2010].

The MOLTO Phrasebook

K. Angelov, O. Caprotti, R. Enache, T. Hallgren, A. Ranta

Chalmers, University of Gothenburg
{krasimir,caprotti,ramona.enache,hallgren,aarne}@chalmers.se

Abstract

This Phrasebook is a program for translating touristic phrases between 14 European languages: Bulgarian, Catalan, Danish, Dutch, English, Finnish, French, German, Italian, Norwegian, Polish, Romanian, Spanish, Swedish. The Phrasebook is implemented in the Grammatical Framework programming language as the first demonstration for the MOLTO EU project (molto-project.eu) and will be extended during the project.

1. Introduction

The MOLTO Phrasebook is a multilingual grammar application developed within the EU MOLTO project to showcase the features of the Grammatical Framework, GF, system. It demonstrates how reliable multilingual translations can be derived from an abstract grammar unifying these translations and allowing to translate from any language to the others. The interlingua used by GF, rather than translating words, focuses on meanings or concepts. The GF programming language combines features from grammar languages to functional programming with categorical grammar formalisms and logical frameworks (Ranta, 2004).

From the programmer's perspective, any GF application builds upon a large library of resource grammars and functors: the GF Resource Grammar Library, that currently makes available programmatic primitives to handle syntax, lexicon and inflection for 22 languages with variable coverage. GF deals with the structural differences between languages at compile time, yielding maximal run-time efficiency. Ideally, leaving the linguistic aspects to the GF libraries, the author of an application grammar needs only basic skills in order to add a new language to an application. In the specific case of the Phrasebook application, many of the grammars were created semi-automatically by generalization from examples and grammar induction from statistical models (Google translate). The various configurations of skills tested during the development of the Phrasebook are presented in Section 3.

GF is distributed for all platforms and GF applications can be compiled to JavaScript making them suitable to the web browsers, irrespective of the device. This possibility alone makes GF a convenient tool for fast prototyping of mobile multilingual applications, such as the MOLTO Phrasebook. From the users' perspective, a GF application can be accessed via a web browser on any device, including mobile phones. Off the shelf JavaScript functions are available to construct a friendly user interface in which allowed word choices guide the selection and/or textual input. Not only does the system use incremental parsing to prompt the possibilities, but it also produces quasi-incremental translations of intermediate results from words or complete sentences. The user interface is presented in Section 4.

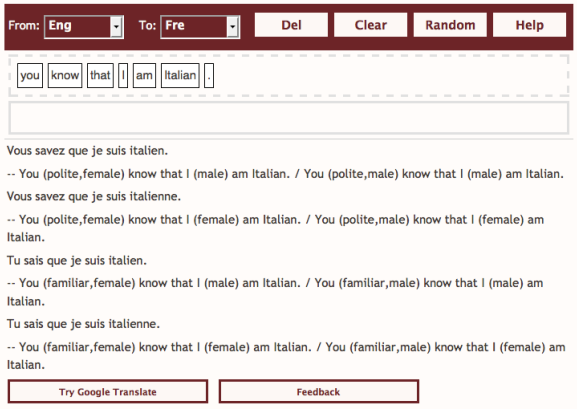


Figure 1: Screen-shot of the online demo

2. Abstract and Concrete Grammars

The GF abstract grammar that captures the object entities and domain of the Phrasebook handles several categories, from units of discourse such as phrases, sentences and questions, to objects like numerable or mass items (*three pizzas* but *some water*), and places, currencies, languages, nationalities, means of transportation, date, and time. It has a collection of constructors that allow to represent for instance a question such as *How far is the zoo?* abstractly as `HowFar(Zoo)` using `HowFar : Place -> Question ; Zoo : PlaceKind`. Each language is produced by linearizing the abstract tree with specific rules that use the GF resource grammar to capture the specific linguistic characteristics. In the example, the French concrete grammar rules are `Zoo = mkPlace (mkN "zoo" masculine) dative` and `HowFar place = mkQS (mkQCl what_distance_IAdv place.name)`. The GF resource grammar for French knows how to build a noun with morphology, `mkN`, a question `mkQS`, and a question clause `mkQCl`. The concrete grammar rule for Swedish is slightly different `HowFar place = mkQS (mkQCl far_IAdv (mkCl (mkVP place.to)))`, yet it is the same as that for Norwegian because of how the resource grammars are designed. Combining it all, the French translation will be *À quelle distance est le zoo?* and the Swedish *Hur långt är det till djurparken?*.

GF application grammars strive for quality. In the

Language	Fluency	GF skills	Informed dev.	Informed testing	Ext. tools	RGL edits	Effort
Bulgarian	***	***	-	-	?	*	**
Catalan	***	***	-	-	?	*	*
Danish	-	***	+	+	**	*	**
Dutch	-	***	+	+	**	*	**
English	**	***	-	+	-	-	*
Finnish	***	***	-	-	?	*	**
French	**	***	-	+	?	*	*
German	*	***	+	+	**	**	***
Italian	***	*	-	-	?	**	**
Norwegian	*	***	+	-	**	*	**
Polish	***	***	+	+	*	*	**
Romanian	***	***	-	-	*	***	***
Spanish	**	*	-	-	?	-	**
Swedish	**	***	-	+	?	-	**

Table 1: Effort estimate

Phrasebook, the kind of quality that can be achieved is exemplified e.g. by sentences that have many translations, each one capturing a flavor of politeness (e.g. "you" in English will have to be disambiguated to polite you, colloquial you and male/female when translating to, say, Italian or French). The abstract grammar makes distinctions between various cases of personal pronouns that identify gender and familiarity, e.g. in greetings or in questions, so that it knows about `IMale` versus `IFemale`, or `YouPolMale` versus `YouFamFemale`. If an ambiguous sentence such as *How old is your daughter?* is entered for translation, it leads to several choices in most languages, for instance in Swedish to *Hur gammal är er dotter?* for the cases of `your(polite,female)` and `your(polite,male)` whereas *Hur gammal är din dotter?* for `your(familiar,female)` and `your(familiar,male)`.

Currently the grammar does not yet cover directions, time and problematic situations, for instance when compared to http://wikitravel.org/en/Phrasebook. With a lexicon of 100 words, the grammar yields 2582 abstract syntax trees of depth 2, which become 656399 of depth 4.

3. The Phrasebook as a Case Study

Developing a multilingual application covering some domain in 14 languages is demanding in terms of language knowledge and quality testing. In Figure 1, we have tracked the type of expertise and effort that was devoted to crafting each single language. Native speakers, fluent in GF and with linguistic background, worked on Bulgarian, Catalan, Polish, and Romanian. However, developers had no knowledge of Danish and Dutch, and had to request the help of native speakers, who were presented with examples generated by a bootstrapped version of the concrete grammars, based on similar languages or on idioms and literal translation taken from the Internet. The full legend for the table is described in (Angelov et al., 2010).

The overall aim is to devise a MOLTO methodology that lowers the cost of adding a new language to a GF application by using automated example-driven grammar generation. The correct design of the batch of examples is language dependent and assumes analysis of the resource grammar, for instance to be able to build inflected words. More precisely, for some languages it is enough to generate examples that show one form of a noun in order to obtain its GF representation (the full inflection table), whereas for other languages, such as German, one has to know up to 6 forms.

4. The Phrasebook at Your Hands

The Phrasebook is distributed as open-source software, licensed under GNU LGPL, from http://code.haskell.org/gf/examples/phrasebook/. It is also available online from the MOLTO project web pages, as a demo and as a mobile application for the Android platform. Users are welcome to send comments, bug reports, and better translation suggestions using the feedback button, as shown in Figure 1. Fall-back to statistical translation is currently implemented just as a link to Google translate, however in future versions, GF will be integrated with tailor-made statistical models.

5. Acknowledgments

The Phrasebook has been built in the MOLTO project funded by the European Commission (FP7/2007-2013) under grant agreement FP7-ICT-247914. The authors are grateful to Inari Listenmaa, Jordi Saludes, and Adam Slaski and to the native speaker informants helping to bootstrap and evaluate the grammars: Richard Bubel, Grégoire Détrez, Rise Eilert, Karin Keijzer, Michał Pałka, Willard Rafnsson, Nick Smallbone.

6. References

K. Angelov, O. Caprotti, R. Enache, T. Hallgren, I. Listenmaa, A. Ranta, J. Saludes, and A. Slaski. 2010. D10.2 molto web service, first version. Project Deliverable D10.2, Gothenburg, Sweden, 06/2010.

A. Ranta. 2004. Grammatical Framework: A Type-Theoretical Grammar Formalism. *The Journal of Functional Programming*, 14(2):145–189.

A Framework for Multilingual Applications on the Android Platform

Grégoire Détrez, Ramona Enache

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
gdetrez@crans.org, ramona.enache@chalmers.se

Abstract

In this paper we describe a Java library allowing applications to use advanced linguistic knowledge on Android devices. We implemented parts of the Grammatical Framework runtime system and optimized it to run on Android handheld devices. This allows building mobile applications on top of GF grammars.

1. Introduction

This paper describes our work in implementing the basic GF runtime system in Java and using it for building applications on the Android platform.

GF (Grammatical Framework) is a type-theoretical grammar formalism and a functional programming language. It is mainly used in multilingual grammar applications for formalizing the syntax of natural languages. Compared to many other approaches to computational linguistics, which are based on statistical methods and machine learning, GF treats natural languages from a programming languages perspective. The key idea of GF is to have an **abstract syntax** defining the main categories and rules that connect them, which is common to all grammars and many **concrete syntaxes** that implement the categories and relations from the abstract syntax, following specific characteristics of the given language. The abstract syntax describes the grammar conceptually and provides a framework for the actual computational grammars, which are the concrete syntax modules. It also limits the coverage of the grammar to the constructions that could be built using the elements of the abstract syntax. From this point of view, GF is similar to other grammar formalisms like HPSG and LFG.

The main operations that can be performed on a GF grammar are **parsing** from natural language to the abstract syntax tree representing the underlying concept and **linearization** that generates natural language constructions in a certain language from an abstract syntax tree. By combining this two operations one obtains a **translation** between any two concrete grammars. This approach has the advantage that the translation will always be syntactically correct, due to the fact that the linearization in a certain grammar, uses the implementation of the concrete syntax module.

In addition to this, GF provides a portable runtime format, PGF (Angelov et al., 2010) which can be used to embed the libraries further on in applications written in programming languages that provide a suitable interpreter. In this way, other projects can use GF modules, as normal software libraries for the development of other projects. PGF interpreters exists for Haskell and JavaScript at the moment, and our work resulted in the Java version of the interpreter.

2. Motivations

There are many motivations to have linguistic applications on handled devices. One can think of automatic translation, tools for languages learner or for travelers and help for impaired people. Many existing services in those categories requires a live connection to the internet, which is not always available, especially when one is traveling abroad.

One of the advantage of GF is its extensive and growing resource library, with formal grammar and basic vocabulary for over 16 languages (Ranta, 2009). The library provides the linguistic background for developing domain-specific grammars and other language applications.

And finally, we choose the Android platform to experiment because of its openness and its growing adoption.

3. Related work

Aarne Ranta implemented an multilingual translator for number working on mobile devices. It was implemented in JavaScript and it ran as a webpage in the device browser. [link please]

The grammatical framework runtime has once been implemented in Java by Björn Bringert (Bringert, 2005) but this implementation was not maintained anymore and didn't follow recent changes in the grammar format and the runtime system.

4. Implementation

The current runtime system being written in Haskell, and since the algorithms for parsing and linearization are specifics to GF, we couldn't use pre-existing libraries and implemented it from scratch.

During the beginning of this project, we concentrated on implementing and optimizing the parser and linearizer. The main reason is that the limited computing power of the targeted devices would make difficult to implement the full GF runtime system.

Those component are enough to build interesting application using natural language. Moreover, for complex grammars, we quickly reach the limits of the devices computing power.

The parsing algorithm is described in (Angelov, 2009) and the linearization algorithm in (Angelov and Ranta, 2010).

5. Application

We developed a simple phrasebook application to demonstrate a possible use of the library (`http://www.grammaticalframework.org/android/`). The application allows the user to enter simple sentences in a controlled language and to translate them in a different language. This application is based on the MOLTO phrasebook project (`http://www.molto-project.eu/demo/phrasebook`). This is a relevant use case as it has a clear potential for usage because of the high quality of the translations and the variety of languages for which the grammar was deviced. It is also worth mentioning that the reasonable coverage of the grammar makes the phrasebook applicable in many day-to-day situations for tourists traveling abroard.

To allow easy and fast input while restraining the user to the controlled language, we used an interface similar to the fridge magnets application (`http://tournesol.cs.chalmers.se:41296/fridge`). This demonstrate the utility of predictive parsing on the cell phone. This feature is a great aid for users of a controlled language, since they can always be aware of the coverage, and the possibilities that the grammar offers. (See screenshot in figure 1.)

What is more is that the Android platform provides services for high-quality voice synthesis for a number of languages, which can be plugged to the grammar applications. This gives our approach a great advantage over the traditional phrasebooks.

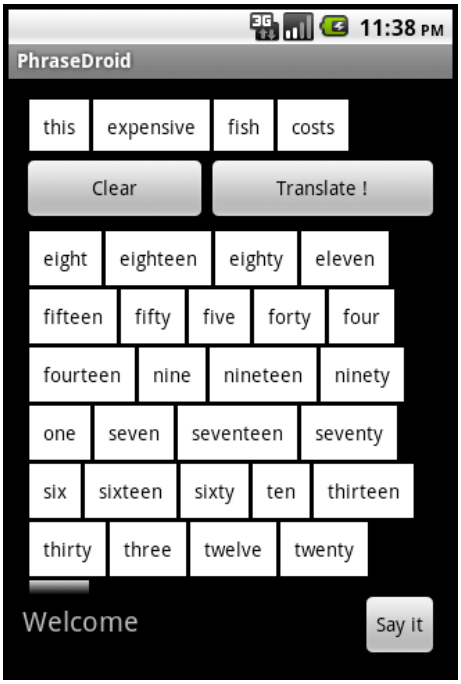


Figure 1: Phrasedroid screenshot

6. Future work

Though we already worked hard on improving the initial performances and to make the user experience acceptable, gain in this domain are still possible. A next step might also be to implement some parts of the logical framework.

And one of our main priority is to keep the library up to date regarding future changes in the GF runtime system.

7. Acknowledgments

We would like to thank Krasimir Angelov for his explanation of the GF algorithms.

8. References

Krasimir Angelov and Aarne Ranta. 2010. Loosely coupled synchronous parallel multiple context-free grammars for machine translation.

Krasimir Angelov, Björn Bringert, and Aarne Ranta. 2010. Pgf: A portable run-time format for type-theoretical grammars. *J. of Logic, Lang. and Inf.*, 19(2):201–228.

Krasimir Angelov. 2009. Incremental parsing with parallel multiple context-free grammars. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 69–76, Morristown, NJ, USA. Association for Computational Linguistics.

Björn Bringert. 2005. Embedded grammars. Master's thesis, Chalmers University of Technology, Göteborg, Sweden, February.

A. Ranta. 2004. Grammatical Framework: A Type-Theoretical Grammar Formalism. *The Journal of Functional Programming*, 14(2):145–189.

Aarne Ranta. 2009. The GF Resource Grammar Library. *Linguistic Issues in Language Technology*, 2.

Revision of Part-of-Speech Tagging in Stockholm Umeå Corpus 2.0

Eva Forsbom*, Kenneth Wilhelmsson†

*Department of Linguistics and Philology
Uppsala University
evafo@stp.lingfil.uu.se

†Swedish School of Library and Information Science
University of Borås
kenneth.wilhelmsson@hb.se

Abstract

Many parsers use a part-of-speech tagger as a first step in parsing. The accuracy of the tagger naturally affects the performance of the parser. In this experiment, we revise 1500+ proposed errors in SUC 2.0 that were mainly found during work with schema parsing, and evaluate tagger instances trained on the revised corpus. The revisions turned out to be beneficial also for the taggers.

1. Introduction

Many parsers of today rely on a statistical part-of-speech tagger as a preprocessing step, in order to rank or limit the amount of possible analyses for each word. However, the tagger is only as good as the data it is trained on, and could potentially be a bottleneck for the correctness of parser systems. If the data contain errors and inconsistencies, the tag distribution for the affected words and n-grams would be skewed. Some of the errors are likely to harm both tagging and parsing (e.g. sentence initial errors), while others may only harm one of the two.

In this paper, we present an initial attempt to investigate if, and how much, tagging accuracy can be enhanced through revising a set of 1500+ potential errors mainly collected in the work concerning schema parsing (Wilhelmsson, 2010) with the Swedish Stockholm-Umeå corpus (Ejerhed et al., 2006). The corpus, henceforth SUC, has become the de facto standard for training and evaluating part-of-speech taggers, as its annotation has been manually revised, and also improved for version 2.0. It still includes errors and inconsistencies, however.

2. Set of changes

The proposed set of changes particularly includes types with severe consequences for parsing, such as tagging of/to verbs, and tagging of/to the markers of sub-clauses or relative clauses. In SUC 2.0, there are five such markers: subjunction, interrogative/relative pronoun, interrogative/relative adverb, interrogative/relative determiner and interrogative/relative possessive.

The following is a typical example of how *som* should be changed from conjunction (KN) to interrogative/relative pronoun (HP) to signal the start of a relative clause:

```
Vad är det som/KN har hänt        (kk27-057)
What is it that   has happened
```

In the graphical user interface of the schema parser, these types of errors yield analyses that are often visually recognizable directly. On the other hand, possible errors concerning more subtle aspects, e.g. gender agreement in NP-chunks, have not been detected to the same extent, as the parser is robust enough to ignore these.

The set of suggested changes affects 2% of the sentences in SUC. The changes are not claimed to reflect the proportions of all the actual errors in SUC 2.0. It is unknown how many these are, what their exact distribution is, and what would be the accuracy for a tagger trained on a corrected, or perfect, corpus.

As it seemed likely that some changes, although linguistically well-motivated, actually would decrease the accuracy, we divided the errors into nine groups (see Table 1). If any of the groups should decrease the accuracy, these groups could be skipped, or the sentences affected could be removed from the training data to increase overall accuracy.

The division was based on error type, with the extra constraints that the number of changes in each group should be large enough to be able to yield significant changes in accuracy score and that the groups should not overlap. Members that could belong to more than one group were therefore placed in the group with the lowest group number. Each group contains 4–15% of the suggested changes.

3. Evaluation

The error groups were evaluated using the statistical TnT tagger (Brants, 2000) and 10-fold cross validation on SUC for three tagsets, as the granularity of the tagset affects tagger performance.

The SUC tagset consists of 150+ tags, but a better tagging accuracy can be achieved with the Granska tagset (Carlberger and Kann, 1999), which is a variation of the SUC tagset, or "Granskaish", which, in turn, is a subset of the Granska tagset that can be mapped back to the SUC tagset losslessly (Forsbom, 2008). The Granska tagset was altered to fit the needs of the Granska grammar checker, adding some features to the tags, and conflating tags with infrequent features. Granskaish only added features for copulas, auxiliaries, singulars (cardinal), and dates.

For each error group, we performed the changes, divided the corpus into 10 partitions, trained a tagging model

Group	Description	No. of changes	SUC	Granska	Granskaish
	No changes	0	95.52±0.15	95.69±0.15	95.62±0.14
	All changes	1569	**95.58±0.15**	**95.74±0.15**	**95.67±0.15**
1	Sentence initial changes	192	**95.53±0.15**	**95.70±0.15**	95.62±0.14
2	Changes to interrogative/rel. adverb	258	95.52±0.15	95.69±0.15	95.62±0.14
3	*Som* to conjunction	92	**95.53±0.15**	**95.70±0.15**	95.62±0.14
4	*Som* to interrogative/rel. pronoun	111	**95.53±0.15**	95.69±0.14	95.62±0.14
5	Changes to conjunction	71	95.52±0.15	95.69±0.15	95.62±0.14
6	Changes to subjunction	130	95.52±0.15	*95.68±0.15*	*95.61±0.14*
7	Changes to adverb	285	**95.55±0.15**	**95.71±0.15**	**95.64±0.15**
8	Changes to preposition	193	**95.53±0.15**	95.69±0.15	95.62±0.15
9	Other changes	237	**95.53±0.15**	**95.70±0.15**	**95.63±0.14**

Table 1: Error groups with overall tagging accuracy and standard deviation.

for each partition and tagset, and ran the 10-fold cross-validation test (see Table 1).

Altogether, the changes improved tagging accuracy, albeit with a small increase in standard deviation for the Granskaish tagset. Group 7 improved the accuracy most, while group 2 and 5 had no effect at all. Group 6 actually decreased the accuracy, at least for the Granska-based tagsets. All other groups had a minor positive effect.

However small, the improvements in accuracy were all statistically significant ($\alpha = 0.001$) using the McNemar test (McNemar, 1947).

4. Discussion

This initial experiment showed that part-of-speech errors that cause problems for a parser are troublesome also for statistical part-of-speech taggers. By revising such errors in the training data, it is possible to improve the accuracy of the tagger, and, most likely, consequently the accuracy of the parser.

Contrary to the initial hypothesis, no group of changes was obviously harmful for all tagsets, although some groups did not improve accuracy. It may still be the case, however, that individual errors in a group actually decrease accuracy.

A natural second step would be to study in more detail how the taggers tag the changed occurrence and its nearest context, and to try to find more errors in a systematic way, e.g. using the variation n-gram method proposed by Dickinson (2005).

5. References

Thorsten Brants. 2000. TnT - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*, Seattle, Washington.

Johan Carlberger and Viggo Kann. 1999. Implementing an efficient part-of-speech tagger. *Software Practice and Experience*, 29(9).

Markus Dickinson. 2005. *Error Detection and Correction in Annotated Corpora*. Ph.D. thesis, Department of Linguistics, The Ohio State University.

Eva Ejerhed, Gunnel Källgren, and Benny Brodda. 2006. Stockholm-Umeå corpus version 2.0. Stockholm University, Department of Linguistics and Umeå University, Department of Linguistics.

Eva Forsbom. 2008. Good tag hunting: Tagability of Granska tags. In Joakim Nivre, Mats Dahllöf, and Beáta Megyesi, editors, *Resourceful Language Technology: Festschrift in Honor of Anna Sågvall Hein*, Acta Universitatis Upsaliensis: Studia Linguistica Upsaliensa 7, pages 77–85. Uppsala University, Uppsala, June.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, June.

Kenneth Wilhelmsson. 2010. *Heuristisk analys med Diderichsens satsschema. Tillämpningar för svensk text [Heuristic Analysis with Diderichsen's Sentence Schema - Applications for Swedish Text]*. Ph.D. thesis, Department of Philosophy, Linguistics and Theory of Science, University of Gothenburg.

Text Summarization using Random Indexing and PageRank

Pär Gustavsson, Arne Jönsson

Department of Computer and Information Science, Santa Anna IT Research Institute AB
Linköping University, SE-581 83, LINKÖPING, SWEDEN
pargu814@student.liu.se, arnjo@ida.liu.se

Abstract

We present results from evaluations of an automatic text summarization technique that uses a combination of Random Indexing and PageRank. In our experiments we use two types of texts: news paper texts and government texts. Our results show that text type as well as other aspects of texts of the same type influence the performance. Combining PageRank and Random Indexing provides the best results on government texts. Adapting a text summarizer for a particular genre can improve text summarization.

1. Introduction

CogSum (Jönsson et al., 2008) is a tool for creating extraction based text summaries based on the vector space technique Random Indexing. To further improve sentence ranking CogSum also uses PageRank (Brin and Page, 1998). To use PageRank we create a graph where a vertex depicts a sentence in the current text and an edge between two different vertices is assigned a weight that depicts how similar these are, by a cosine angle comparison. Sentences with similar content will then contribute with positive support to each other. This effect doesn't exclusively depend on the number of sentences supporting a sentence, but also on the rank of the linking sentences. This means that a few high-ranked sentences provide bigger support than a large number of low-ranked sentences. This leads to a ranking of the sentences by their importance to the document at hand and thus to a summary including only the most important sentences.

2. Experiment

To evaluate CogSum for text summarization on various text types, two studies were performed. The first compared summaries created by CogSum with or without PageRank activated. This study was conducted on news texts and we used another summarizer, SweSum (Dalianis, 2000), as a baseline. SweSum is basically a word based summarizer but with additional features such as letting users add keywords, extracting abbreviations and having a morphological analysis. SweSum has been been tailored to news texts in various ways, e.g. by increasing the probability to include the first sentences in an article in the summary.

The created summaries were compared to existing gold standards in the KTH eXtract Corpus (KTHxc) by an overlap measure on sentence level (Hassel and Dalianis, 2005). We used 10 Swedish news texts with an average length of 338 words.

The second study was conducted to compare summaries created by the same systems but with other texts, namely 5 fact sheets from the the Swedish Social Insurance Administration (Sw. Försäkringskassan). The length of the fact sheets ranged from 1000 to 1300 words. The gold standards for these texts were created by Carlsson (2009). The evaluation for this experiment was conducted in AutoSummENG, by means of the metric Graph Value Similarity (Giannakopoulos et al., 2008), as this allows taking content similarity between different sentences into consideration during the evaluation.

The Random Indexing dimensionality was kept constant to 100 through the first experiment, as done previously by Chatterjee and Mohan (2007) on texts of equal length. Different dimensionalities ranging from 100 to 1000 were initially used in the second study as these texts were longer on average. The summaries created in the second study were more or less identical, especially the ones with a dimensionality of 500 and upwards. Results from previous studies imply that as low dimensionality as possible is desirable to deal with time and memory usage while it's unimportant to optimize the variable because of the small difference between the created summaries (Sjöbergh, 2006). With this in mind a dimensionality of 500 was used for the second study.

3. Results

Text	CogSum	CogSumPR	SweSum
Text001	85.71	85.71	85.71
Text002	30.00	9.09	38.10
Text003	20.00	0.00	80.00
Text004	57.14	54.54	52.63
Text005	70.59	35.29	66.67
Text006	66.67	66.67	50.00
Text007	50.00	50.00	85.71
Text008	42.86	66.67	50.00
Text009	40.00	37.50	70.59
Text010	28.57	33.33	66.67
Average	49.15	43.88	64.61

Table 1: Sentence overlap on news texts (%)

Table 1 shows results from the first study for the summaries created by CogSum with or without PageRank and SweSum for 10 news texts from the KTHxc corpus. The table shows the overlap on sentence level compared to the gold standards expressed in percentage. We can see that SweSum gained the highest average sentence overlap of 64.61% followed by CogSum (49.15%) and CogSumPR (43.88%).

The results from the second study, where we use government texts are presented in Table 2. The table shows the N-gram Value Similarity between the created summaries and

the gold standards. The value of this metric ranges from 0 to 1.

Text	CogSum	CogSumPR	SweSum
Text001	0.532	0.491	0.227
Text002	0.284	0.356	0.353
Text003	0.416	0.443	0.293
Text004	0.292	0.383	0.168
Text005	0.370	0.342	0.246
Average	0.379	0.403	0.258

Table 2: Graph Value Similarity on government texts

As shown in Table 2 the summaries created by CogSumPR gained the highest average value of 0.403 followed by CogSum (0.379) and SweSum (0.258).

To further investigate the various evaluation metrics used in our study, we evaluated the news paper texts, i.e. the first experiment, using AutoSummENG.

Graph Value	CogSum	CogSumPR	SweSum
Average	0.526	0.516	0.584

Table 3: Graph Value Similarity on news texts

Table 3 presents the results, and as can be seen they are consistent with the first study as the systems get ranked in the same order as they did when ranked according to sentence overlap, c.f. Table 1.

4. Discussion

The results of the first study showed that SweSum achieved the best results. This is not surprising as this system is tailored to summarize news texts. The results for CogSum and CogSumPR were equal for most of the texts in the corpus with a slight advantage for CogSum. One particularly interesting result is the one for Text003 where SweSum got an 80% overlap while CogSum gained 20% and CogSumPR 0%, which call for further analysis in the future to be properly explained. It was hard to draw any definite conclusions from this data and the possibility that CogSum performed better than CogSumPR by chance exists. Still, it's possible that Random Indexing works well as it is and that the incorporation of a PageRank algorithm doesn't improve the created summaries.

The second study revealed that the summaries created by CogSum with PageRank activated were closest to the gold standards which means that they were created by a better system. This is only the case for the 5 texts used in this study and a larger evaluation would strengthen the reliability of the study. The results showed that CogSum with and without PageRank performed relatively equal results for all of the texts which indicates that the two systems gained an accurate "understanding" of all of them. The fact that the activation of PageRank led to a better average result for these five fact sheets thus suggest that this version of the summarizer may be preferable for this kind of texts in general. No statistical significance testing was conducted in either study due to the fairly small number of texts used, but further studies involving a larger amount of texts are close at hand.

One possible explanation to the results could be properties of the texts. The fact sheets were longer than the news texts. It is possible that PageRank works better for texts with more sentences as a larger number of sentences can be used to strengthen the mutual effect. Another possible explanation is the structure of the texts used in the two studies. The fact sheets aim to contribute with as much information as possible regarding a certain topic and thus have a fair number of headings. The news texts on the other hand only include a main header and read up on a news item with the most important information presented in the beginning of the text.

The evaluations were done automatically with no qualitative input from people in potential target groups. Although humans were involved in the creation of the gold standards and thus affected the results indirectly, no information regarding readability or usefulness of the summaries were collected. The results only show how different extraction techniques mimic human choice of extraction units.

Acknowledgment

This research is financed by Santa Anna IT Research Institute AB.

5. References

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117.

Bertil Carlsson. 2009. Guldstandarder – dess skapande och utvärdering. Master's thesis, Linköping University.

Nilhadri Chatterjee and Shiwali Mohan. 2007. Extraction-based single-document summarization using random indexing. In *Proceedings of the 19th IEEE international Conference on Tools with Artificial intelligence – (ICTAI 2007)*, pages 448–455.

Hercules Dalianis. 2000. Swesum – a text summarizer for swedish. Technical Report TRITA-NA-P0015, IPLab-174, NADA, KTH, Sweden.

George Giannakopoulos, Vangelis Karkaletsis, George Vouros, and Panagiotis Stamatopoulos. 2008. Summarization system evaluation revisited: N-gram graphs. *ACM Transactions on Speech Language Processing*, 5(3):1–39.

Martin Hassel and Hercules Dalianis. 2005. Generation of Reference Summaries. In *Proceedings of 2nd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, Poznan, Poland, April 21-23.

Arne Jönsson, Mimi Axelsson, Erica Bergenholm, Bertil Carlsson, Gro Dahlbom, Pär Gustavsson, Jonas Rybing, and Christian Smith. 2008. Skim reading of audio information. In *Proceedings of the The second Swedish Language Technology Conference (SLTC-08), Stockholm, Sweden*.

Jonas Sjöbergh. 2006. *Language Technology for the Lazy - Avoiding Work by Using Statistics and Machine Learning*. Ph.D. thesis, KTH, Stockholm, Sweden.

Repair-transitions in transition-based parsing

Martin Haulrich

Dept. of International Language Studies and Computational Linguistics
Copenhagen Business School
mwh.isv@cbs.dk

1. Introduction

Transition-based parsing has been shown to yield state-of-the-art results in dependency parsing thanks to deterministic processing with very rich feature models. One of the drawbacks of transition-based parsing is its greedy nature. If an incorrect decision is made, this cannot be changed. Furthermore, future decisions are based on the result of this incorrect decision, which can then lead to error propagation (McDonald and Nivre, 2007).

In this work, we introduce a repair-transition that allows the parser to remove a previously added dependency arc from the analysis. We analyze how to best train a parser with this transition and show that this method leads to better parsing accuracy on English data compared to a standard transition-based parser.

2. Transition-based dependency parsing

The core of a transition-based parser is a parsing algorithm consisting of a transition system and an oracle (Nivre, 2008). The oracle is used during training to determine a transition sequence that leads to the correct parse. From these oracle transition sequences a model is trained to predict which transition should be used during parsing.

A number of different parsing algorithms exist. Here we will focus on the one called *NivreEager*. This algorithm uses two data structures, a stack of partially analyzed word tokens and a buffer of remaining input tokens, and the following four transitions:

Shift Push the token at the head of the buffer onto the stack.

Reduce Pop the token on the top of the stack.

Left-Arc$_l$ Add to the analysis an arc with label l from the token at the head of the buffer to the token on the top of the stack, and push the buffer-token onto the stack.

Right-Arc$_l$ Add to the analysis an arc with label l from the token on the top of the stack to the token at the head of the buffer, and pop the stack.

3. Repair

Transition-based parsers are greedy, and this can lead to errors in parsing. Figure 1 shows a sentence where a standard transition-based parser makes a greedy choice that is incorrect. When the parser encounters the word *believes* it chooses to make this the root of the sentence, which is not correct - the conjunction *and* should be the root. When the parser encounters *and*, it has already chosen *believes* as the root and cannot change this decision. Apart from *believes* being analyzed incorrectly, the decision also leads to *and* and '.' being analyzed incorrectly. This is what is called error propagation.

This problem motivates the use of repair-transitions. Repair transitions are transitions that can *repair* the errors made by the parser. Here we focus on one repair-transition:

Remove-ra-d Remove the incoming arc on the token at the top of the stack.

In the sentence in Figure 1, this means that in a state where *believes* is at the top of the stack, the parser can choose the *remove-ra-d* transition and remove the ROOT-arc from the $<ROOT>$-token.

3.1 Parsing

The transition-based parser with the *remove-ra-d* repair transition introduced above first checks if the repair-transition should be used. If so, it applies the transition. If not, the parser performs a non-repair transition as usual. This means that the parser has two models. One standard-parsing model and one repair model.

3.2 Training

To train the repair-model for the parser an oracle that can tell when the parser makes mistakes is needed. This oracle is created by using a standard parser on gold-standard data and seeing when the parser makes mistakes.

We first train a standard parser without repair-transitions. We then use this parser to parse gold-standard texts. During the parsing, situations where the repair-transition should be used, can be identified. These are situations where the token at the top of the stack has a head that is different from the head it has in the gold-standard (or a different relation).

When states where the repair-transition should be applied have been identified, a classifier can be trained to predict in a given state whether or not the repair-transition should be used. This is the repair model.

4. Experiments

4.1 Software

All experiments have been performed using MaltParser (Nivre et al., 2006) (v. 1.3.1). We have extended this with the use of the repair-transition as described above. In all experiments the same features and parameters have been used for the two models in the parser.

4.2 Data

We have used the English data from CoNLL-07 shared task (Nivre et al., 2007). The training data consists of 400.000

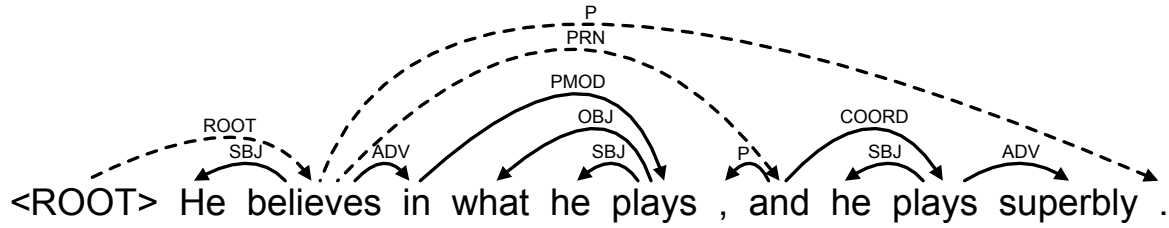


Figure 1: Example of sentence parsed with standard parser. Dotted arcs are incorrect.

tokens in 16.000 sentences and the test data consists of 5.000 tokens in just above 200 sentences. The parameters and features used for the parser are those used by the MaltParser in the CoNLL-07 shared task.

4.3 Training regime

Given that we have a limited amount of training data, an important question to answer is how to use the data as we are actually training two different models. A standard parsing model and the repair model. The repair-parser should be able to correct errors that the standard parser makes on unseen data, so the obvious choice would be to reserve some of the training data for the training of the repair model and use only this data for training this model.

To test this hypothesis we have split the training data into two parts, A and B. We have trained three repair parsers with three different repair models. In all of them the standard model is trained on A. One repair model has been trained on A, i.e. only on data seen by the standard model. One repair model has been trained on B, i.e. only on data *not* seen by the standard model. The last repair model has been trained on A and B, i.e. a mix of seen and unseen data.

Std. model	Rep. model	LAS
A		86.04
A	A	86.83
A	B	84.60
A	A+B	86.97

Table 1: LAS for one standard parser and three repair parsers. The first column shows the part of the training data used for the standard model. The second column shows the part of the training data used for the repair model.

Table 1 shows the results of these experiments. The hypothesis that the repair model should be trained on unseen data, seems to be incorrect. If the repair model is trained only on unseen data, the accuracy of the parser decreases compared to the standard parser. If it is trained only on seen data the accuracy increases. The results for the last model (A+B) shows that unseen data does not necessarily decrease the performance - as long as the model is also trained with seen data.

4.4 Results

Table 2 shows the final results on evaluation data. The repair parser achieves significantly ($p < 0.01$) higher accuracy than the standard parser.

	Standard	Repair
LAS	86.33	†**87.48**
UAS	87.41	†**88.50**
LA	89.14	†**90.41**

Table 2: Results on CoNLL-07 shared task evaluation data with standard parser and parser with repair-transition.

5. Conclusion

We have shown how to define, use and train repair-transitions in transition-based parsing. We have also shown that on at least one data set the new parser leads to significant improvements in accuracy compared to traditional transition-based parsing. To achieve this improvement, it is vital that the training data used for the standard model in the parser is also used for training the repair model.

6. Future work

In the experiments here the repair-model uses the same features and parameters as the standard model. Higher parsing accuracy can probably be achieved by doing feature selection and parameter optimization for the repair model.

We have only worked with one repair transition, *remove-ra-d*. Other repair-transitions can be defined, and this is something we will work with in the future.

7. Acknowledgments

Thanks to Johan Hall for help with implementation and evaluation, and to Joakim Nivre for numerous useful comments.

8. References

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the EMNLP-CoNLL 2007*, pages 122–131.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of the LREC 2006*, pages 2216–2219, May.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.

Automatic Annotation of Bibliographical References for Descriptive Language Materials

Harald Hammarström

Radboud Universiteit Nijmegen and Max Planck Institute for Evolutionary Anthropology
Department of Linguistics, Deutscher Platz 6, 04103 Leipzig, Germany
h.hammarstrom@let.ru.nl

Abstract

Bibliographical references can be seen as short snippets of natural language text. Searching and browsing bibliographical references are thus instances of Information Retrieval problems. In the present paper, we discuss a particular collection of some 180 000 bibliographical references to descriptive materials of the language of the world. The end user community especially requests the references to be annotated with labels describing the identity of the content (e.g., a particular language) and type of content (e.g., a dictionary or a grammar) of the reference. Since part of the collection is already annotated with such labels, the problem is to devise a supervised learner ("labeler") that can accurately label the unlabeled references, using the labeled ones as training data. Given the specific structure of the problem domain, namely that a) documents are short, b) documents can be written in a wide variety of different languages, c) labels can be signalled through existence/non-existence of a few trigger words, d) some labels are common while other labels are very rare, we suggest an approach based on searching for short DNF boolean formulae (similar to, but preferable to, Decision Trees).

1. Introduction

LangDoc is a large-scale project to list bibliographical references to descriptive materials to all of the ca 7 000 languages of the world (Hammarström and Nordhoff, 2010). The present collection contains about 180 000 such references.

A linguist, typically a typologist searching/browsing through references, would want the collection *systematically* annotated with metadata, such as the identity of the (target-)language(s) the reference treats, the geographical location country/continent, the content-type of the document the reference refers to (e.g., (full-length) `grammar`, `grammar sketch`, `dictionary`, `phonological description`) and so on.

The present collection of 180 000 references comes from a variety of sources, some of which are already annotated with metadata, and this can be exploited in terms of supervised learning.

For example, a bibliographical reference to a descriptive work may look as follows:

> Schneider, Joseph. 1962. *Grammatik der Sulka-Sprache (Neubritannien)* (Micro-Bibloteca Anthropos 36). Posieux: Anthropos Institut.

This reference happens to describe a Papuan language called `Sulka [sua]`, it is a `grammar` (rather than a `dictionary`, `grammar sketch` etc.), and is further tagged with `Oceania` (continent) and `Papua New Guinea` (country). This example reference is written in German (i.e., the (meta-)language that the publication, and therefore reference, is written in – not the (target-)language that the publication aims to describe). The collection as a whole spans some 29 (meta-)languages.

Now suppose we are given a new bibliographical reference which has no annotation. We would like to automatically annotate it with identity, type and whatever other labels are justified, given the training data consisting of already annotated references. For example, many titles in the training data will contain the word "Grammatik" and be annotated with `grammar`, those few which have the word "Neubritannien" will likely be annotated with `Oceania` and `Papua New Guinea` and so on.

At first, this problem, i.e., reference annotation by keyword triggers, might seem like a very easy problem – just find title words which are statistically overrepresented with an annotation label in the training data, and then label new instances as such words occur in their titles. However, there are a few reasons why it is not that simple. A label may be signalled by more than one word, e.g., "kurzgefaßte grammatik" signals `grammar sketch` rather than `grammar` (not both!). It is not given which keyword(s) signal which label(s), e.g., from the example above, is it "Grammatik", "der" or "Grammatik der" (all of them statistically significant) that signals `grammar`? Some labels are very common (and thus have frequent trigger words) while other labels are very uncommon (and thus their trigger words are very uncommon). Typically, a small set of trigger words "account" for an annotation label, i.e., no single one of them has a high recall with its label, but together they do. For example, among 15 236 references annotated for content-type 19 921 distinct word types are present. 3 220 have the label `grammar` and 6 have the label `Sulka [sua]`.

`grammar`

	contain	# overlap	precision	recall
162	"grammatik"	91	0.56	0.068
668	"der"	137	0.21	0.103
84	"grammatik", "der"	48	0.57	0.036
1	"sulka"	1	1.00	0.001

`Sulka [sua]`

	contain	# overlap	precision	recall
1	"sulka"	1	1.00	0.16
668	"der"	4	0.01	0.67

2. A DNF Approach

As outlined in the problem description, our domain knowledge suggests that a label can be inferred if and only if a suitable combination of words is present/absent in a given publication title. More formally:

- A trigger-signature $t = w_1 \wedge \ldots \wedge w_k \wedge \ldots \neg w_{k+1} \wedge \ldots \wedge \neg w_{k'}$ for a label l is a conjunct formula of negated/un-negated terms, such that if a title contains all the un-negated terms but none of the negated terms, then the label l should be inferred
- Each label l can have one or more trigger-signatures $t_1, \ldots, t_n$

For example, one trigger for the label `grammar` might be $\{grammar, \neg sketch\}$, and the full set of triggers for `grammar` might contain $\{grammar, \neg sketch\}$, $\{grammaire\}$, $\{complete, description\}$, $\{phonologie, morphologie, syntax\}$ and so on. Since titles are short (less than 20 words or so), we envisage triggers to be short.

In other words, a classifier (one for each label) can be described as a boolean formula in DNF, where each disjunct corresponds to a trigger. Moreover, each disjunct can be expected to be relatively short.

Thus, all we need to do is to search for a formula in DNF form which can be expected to have only short disjuncts and which is preferably short (in its number of disjuncts). Thus, a simple algorithm is to start from an empty formula and build it larger as accuracy increases with respect to a label in the training data. One can build a formula larger i) by adding a negated/un-negated term to one of its disjuncts (replacing that disjunct), or, ii) by adding a negated/un-negated term to one of its disjuncts (keeping both an updated and un-updated disjunct), or, iii) by adding a new disjunct, inhabited by a negated/un-negated literal. Since we are interested in both high precision and high recall, a natural way to measure accuracy is f-score. Formally:
$d_i \subseteq \Sigma^*$ be a document, i.e., a set of strings
$D = \{d_1, \ldots, d_n\}$ be a set of documents
$W_D = \bigcup d_i$ be the set of terms of a set of documents
$L_D(l) = \{i | d_i \text{ has label } l\}$ be the subset of documents with label l
$c = \bigvee t_j$ be a DNF boolean formula
$c_D = \{i | c \text{ is true for } d_i\}$ be the subset of documents whose terms satisfy a boolean formula c
$Precision_D(c, l) = |c_D \cap L_D(l)|/|c_D|$
$Recall_D(c, l) = |c_D \cap L_D(l)|/|L_D(l)|$
The training algorithm can be described as follows:

1. Start with a label l, a document collection D and an empty formula c
2. Form sets of candicate formulae

$$C' = \{c \vee w | w \in W_D\} \cup \{c \vee \neg w | w \in W_D\}$$
$$C'' = \{ins(w, t_j, c) | w \in W_D, t_j \text{ of } c\} \cup \{ins(\neg w, t_j, c) | w \in W_D, t_j \text{ of } c\}$$
$$C''' = \{ins(w, t_j, c) \vee t_j | w \in W_D, t_j \text{ of } c\} \cup \{ins(\neg w, t_j, c) \vee t_j | w \in W_D, t_j \text{ of } c\}$$

   where $ins(x, t_j, c)$ means "replace t_j with $t_j \wedge x$ in the formula c", e.g., $ins(c, t_2, (a \wedge \neg b) \vee (a)) = (a \wedge \neg b) \vee (a \wedge c)$.
3. Compute $c' = argmax_{c' \in C' \cup C'' \cup C'''}$f-score$_D(c', l)$
4. If $c' == c$ finish, otherwise jump to step 2

3. Results and Discussion

Classifiers for some 3 000 different labels were trained. Nearly all of these labels are uncommon and get short formulae with high (> 0.75) f-score. The common labels get f-scores in the range 0.5-1.0, nearly all trigger-signatures are short, but the length of the DNF may exceed 100 disjuncts. This is significantly better than Decision Trees (Quinlan, 1986) whose performance on this problem (with one tree per label) yields much larger trees for the same f-scores, and requires threshold (tree-height) settings for training to stop.

The output formulae are readily interpretable to a human, thus the classifier annotating a new reference can "explain" its result. Different disjuncts within one formula can be interpreted as cross-language and (intra-language) translation equivalents, e.g., $morphosyntax \vee (grammar \wedge \neg sketch) \vee grammaire \vee grammatik \vee grammatika \vee langue \vee arte \vee course \vee handbook \vee spraakkunst \vee structure \vee grammatica \vee \ldots$.

Training the classifier is slow, given the search space with a large W_D. It is likely that intelligent filtering of W_D may significantly reduce it, but since training speed is not an issue, this has not been explored.

The approach in the present paper generalizes the method of (Hammarström, 2008) to annotate bibliographical references with only uncommon labels. We are not aware of any other work specifically targeting the annotation of bibliographical references. Neither are we aware of related work on a domain with different document content but with a similar structure, i.e., short documents, many languages etc., but given the generality of such a domain, presumably, such work exists.

4. Conclusion

We have shown how to train a high-accuracy short-document label-annotator that a) can handle multiword triggers elegantly b) finds rare as well as common trigger words c) allows "combining" medium-recall triggers into high recall, thus distinguishing them from spurious medium-recall words like "der" or "of", and d) is not likely to overfit.

5. References

Harald Hammarström and Sebastian Nordhoff. 2010. Langdoc: Bibliographic infrastructure for linguistic typology. *Oslo Studies in Language*, to appear:to appear.

Harald Hammarström. 2008. Automatic annotation of bibliographical references with target language. In *Proceedings of MMIES-2: Wokshop on Multi-source, Multilingual Information Extraction and Summarization*, pages 57–64. ACL.

J. R. Quinlan. 1986. Induction of decision trees. *Machine Learning*, 1(1):81–106.

Methods for human evaluation of machine translation

Sofia Bremin†, Hongzhan Hu†, Johanna Karlsson†, Anna Prytz Lillkull†, Martin Wester†, Henrik Danielsson‡ and Sara Stymne†

†Department of Computer and Information Science, ‡Department of Behavioural Sciences and Learning
Linköping University, 58183 Linköping, Sweden
{sofbr664,honhu753,johka299,annpr075,marwe844}@student.liu.se,{first_name.last_name}@liu.se

1. Introduction

Evaluation of machine translation (MT) is a difficult task, both for humans, and using automatic metrics. The main difficulty lies in the fact that there is not one single correct translation, but many alternative good translation options. MT systems are often evaluated using automatic metrics, which commonly rely on comparing a translation to only a single human reference translation. An alternative is different types of human evaluations, commonly ranking between systems or estimations of adequacy and fluency on some scale, or error analyses.

We have explored four different evaluation methods on output from three different statistical MT systems. The main focus is on different types of human evaluation. We compare two conventional evaluation methods, human error analysis and automatic metrics, to two lesser used evaluation methods based on reading comprehension and eye-tracking. These two methods of evaluations are performed without the subjects seeing the source sentence.

There have been few previous attempts of using reading comprehension and eye-tracking for MT evaluation. One example of a reading comprehension study is Fuji (1999) who conducted an experiment to compare English-to-Japanese MT to several versions of manual corrections of the system output. He found significant differences between texts with large differences on reading comprehension questions. Doherty and O'Brien (2009) is the only study we are aware of using eye-tracking for MT evaluation. They found that the average gaze time and fixation counts were significantly lower for sentences judged as excellent in an earlier evaluation, than for bad sentences.

Like previous research we find that both reading comprehension and eye-tracking can be useful for MT evaluation. The results of these methods are consistent with the other methods for comparison between systems with a big quality difference. For systems with similar quality, however, the different evaluation methods often does not show any significant differences.

2. MT systems

We applied our evaluation methods to three different English-to-Swedish phrase-based statistical machine translation systems, all built using the Moses toolkit (Koehn et al., 2007) and trained on the Europarl corpus (Koehn, 2005). Two systems differ in the amount of training data, *Large*, with 701,157 sentences, and *Small* with 100,000 sentences. The third system, *Comp*, uses the same training data as *Large*, and additional modules for compound processing (Stymne and Holmqvist, 2008). These systems are also compared to the human reference translation in Europarl.

2.1 Test texts

We performed the evaluation on four short Europarl texts, from the fourth quarter of 1999, which has been reserved for testing. The texts have 504-636 words. All results are aggregated over the four texts.

3. Evaluation

We have explored four types of evaluations: automatic metrics, human error analysis, reading comprehension and eye-tracking. The human error analysis was made by two persons. They had an inter-rater reliability of 87.8% (Kappa: 0.63). The reading comprehension and eye-tracking studies were performed as a user study with 33 subjects for reading comprehension, and 23 of those 33 for eye-tracking. In these studies the subject saw one text each from the three MT systems, and the human translation.

3.1 Automatic metrics

Table 1 shows Bleu (Papineni et al., 2002) and Meteor (Lavie and Agarwal, 2007) scores for the different systems. On both metrics *Small* is significantly worse than the other two systems. The other systems have more similar scores, with no significant differences, but the trend of which system is better is opposite on the two metrics.

	Meteor	**Bleu**
Comp	17.48	58.02
Large	16.96	58.58
Small	14.33	55.67

Table 1: Metric scores

3.2 Human error analysis

A human error analysis was performed, where errors were identified and classified into six error categories, based on Vilar et al. (2006). The result of the error analysis is shown in Figure 1. Overall the interaction between error type and translation type was significant. The *Small* system has the highest number of errors, especially for incorrect words, which is not surprising considering that it is trained on less data than the other systems. *Comp* has significantly fewer errors than *Large*.

3.3 Reading comprehension

A reading comprehension test was performed using a questionnaire based on Fuji (1999) that was distributed after

	Correct answers	Confidence of correct answers	Estimated fluency	Estimated comprehension	Estimated errors
Human	64.50%	7.19	5.56	5.70	2.94
Comp	59.50%	6.43	3.50	4.85	5.67
Large	67.25%	6.82	4.16	4.86	5.34
Small	59.25%	5.97	3.33	4.53	6.11

Table 2: Reading comprehension results

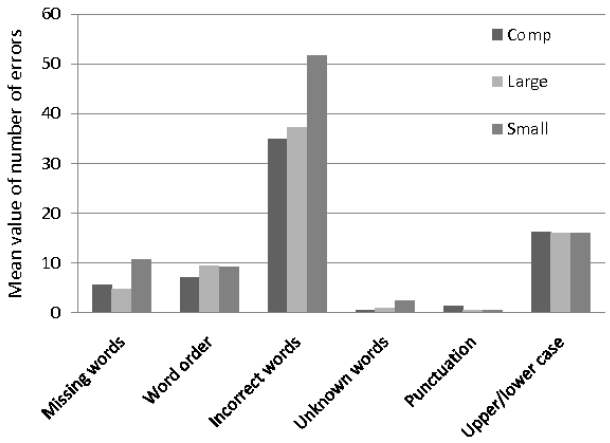


Figure 1: Frequencies of errors

reading each text. The questionnaires contained three content related multiple-choice comprehension questions. The confidence of each answer and three evaluation questions of the readers' impression of the text were rated on a scale from 1-8.

The results on the questionnaires are shown in Table 2. The differences between all systems are not significant. The number of correct answers is actually higher for the *Large* system than for the human reference, but the confidence of the correct answers is lower. On the estimation questions the human reference is best in all cases, and *Small* worst, with *Large* a bit better than *Comp* in the middle.

3.4 Eye-tracking

The eye-tracking study was performed using a SMI Remote Eye Tracking Device. Error boxes were placed on errors in four of the error categories from the error analysis. Control boxes were put in the beginning, middle and end of each sentence, when there was no error box there. Fixation time and number of fixations were measured for error and control boxes, and for the full text. The error boxes had significantly longer fixation times and a higher number of fixations than the control boxes. We also found that different types of errors had significantly different fixation time, with word order errors having the longest fixations, and untranslated words the shortest. This indicates that some error types are more problematic than others for human readers. The fixation time of error boxes were significantly different between the three MT systems with *Small* having the longest and *Large* the shortest fixation times. The same tendency could be seen for the number of fixations. *Small* had a significantly longer overall fixation time than the human reference. For the other systems there were no significant differences in overall fixation time.

4. Discussion and conclusion

It is hard to tell different MT systems apart on texts that are as short as the ones used in this study. Several of the methods did not give significant differences between the systems. But a trend over all methods is that *Small* is worse than both the other two systems and the human text. For the other two systems though, it is hard to say which is best, with mixed metric results, *Comp* having fewer errors on the error analysis, and *Large* having somewhat better result on the reading comprehension and eye-tracking. More research is needed into making a more fine-grained analysis of the difference between systems of similar quality.

Overall we have shown that reading comprehension and eye-tracking give similar results to other evaluation methods for system with large quality differences. For systems with similar quality, however, the methods do not give consistent results. For such systems we believe it is especially important to know which aspects of the translations that are important for the intended usage of the MT system, and choose an evaluation method that measures that.

5. References

S. Doherty and S. O'Brien. 2009. Can mt output be evaluated through eye tracking? In *Proceedings of MT Summit XII*, pages 214–221, Ottawa, Ontario, Canada.

M. Fuji. 1999. Evaluation experiment for reading comprehension of machine translation outputs. In *Proceedings of MT Summit VII*, pages 285–289, Singapore.

P. Koehn et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL'07, demonstration session*, pages 177–180, Prague, Czech Republic.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit X*, pages 79–86, Phuket, Thailand.

A. Lavie and A. Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of WMT'07*, pages 228–231, Prague, Czech Republic.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL'02*, pages 311–318, Philadelphia, Pennsylvania, USA.

S. Stymne and M. Holmqvist. 2008. Processing of Swedish compounds for phrase-based statistical machine translation. In *Proceedings of EAMT'08*, pages 180–189, Hamburg, Germany.

D. Vilar, J. Xu, L. F. D'Haro, and H. Ney. 2006. Error analysis of machine translation output. In *Proceedings of LREC'06*, pages 697–702, Genoa, Italy, May.

Let's MT! — A Platform for Sharing SMT Training Data

Jörg Tiedemann, Per Weijnitz

Department of Linguistics and Philology
Uppsala University
jorg.tiedemann@lingfil.uu.se, per.weijnitz@convertus.se

Abstract

In this paper we describe the LetsMT! platform for sharing training data for building user-specific machine translation models. We give an overview of the general structure of the data repository including the flexible internal storage format that will be used to access data via a transparent user interface. Several tools will be integrated in the platform that support not only uploading data in various formats but also the verification, conversion and alignment of translated documents. The shared resources can then be used within the platform to train tailored translation models using existing state-of-the-art technology that we will integrate in LetsMT! In this paper we show the potentials of such an approach by comparing a domain-specific system with the general purpose engine provided by Google Translate. Our results suggest that domain-specific models may lead to substantial gains even when trained on scarce resources.

1. Introduction

In recent years, statistical machine translation (SMT) has become the leading paradigm for machine translation. However, the quality of SMT systems largely depends on the size and appropriateness of training data. Training SMT models becomes a major challenge for less supported languages since parallel corpora of reasonable size are only available for a few languages. Furthermore, most parallel resources come from very restricted domains and models trained on these collections will always have a strong bias towards the domain of the training data.

To fully exploit the huge potential of existing open SMT technologies we propose to build an innovative online collaborative platform (LetsMT![1]) for data sharing and MT building. This platform will support the upload of public as well as proprietary MT training data allowing users to build multiple MT systems according to their selections of shared training data. Permissions to access uploaded content will be set by the users allowing them to define user groups to share the data with. We will stress the possibility of data privacy that will motivate professional users to use our platform but we hope to achieve a liberal sharing policy among our users.

The main goal of LetsMT! is to make SMT technology accessible for anyone and to enable every-day users to build tailored translation engines on their own and user-contributed data collections without worrying about technical requirements. Initial data sets and baseline systems will be made available to show the potentials of the system and to motivate users to upload and share their resources.

In this paper we describe the general structure of the data repository and the internal storage format that we will use. Finally, we also include a test case illustrating the benefits of domain-specific SMT models compared to general purpose translation using state-of-the art MT provided by Google.

[1]LetsMT! is a ICT PSP PB Pilot Type B project from the area CIP-ICT-PSP.2009.5.1 Multilingual Web: Machine translation for the multilingual web.

2. The LetsMT! Data Repository

One of the key functions of the LetsMT! platform is to provide the possibility to train domain-specific SMT models tailored towards specific needs of its users. For this appropriate data resources are required. LetsMT! is based on data sharing and user collaboration. We will allow data uploads in a variety of formats and store all resources in a unified internal storage format.

The LetsMT! data repository will be based on a robust *version-controlled file system*. We will use a simple and clear file structure to store parallel and monolingual data. Each corpus identified by a unique name (parallel or monolingual) will be stored in a separate version-controlled repository. The name of the corpus will be used as the name of this repository and may contain arbitrary numbers of documents. Repositories can be created by any user but each user will only have access to his/her own branch inside this repository that will be set up during creation time. Each LetsMT! user can then work with a copy of existing corpora through branching (of course only if permissions allow that). In this way we create a space-efficient and flexible environment allowing users to share data and even to apply changes to their copy without breaking data integrity. This will allow us to integrate on-line tools for personal data refinement, for example, tools for adjusting sentence alignments. These refinements can again be shared between users. Another benefit of version-control systems is that changes can be traced back in time. Specific revisions can be retrieved on demand and data releases can be defined.

Inside each repository we will keep the original uploads in their raw format in order to allow roll-back functionalities. Furthermore, pre-processed data in our internal corpus format will be stored together with their meta-data. We will use ISO 639-3 language codes to organize the data collection in appropriate subdirectories. Meta-data will also be stored in a central database allowing users to quickly browse and select training data according to their needs.

Internally all uploaded documents will be converted to a simple XML format which is easy to process and convert. Basically, we will add appropriate sentence boundaries to

the textual contents with unique identifies within each document. Sentence alignments will be stored in separate files with pointers referring to sentences in the corpus. An example is given in figure 1.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE cesAlign PUBLIC "-//CES//DTD XML
          cesAlign//EN" "">
<cesAlign version="1.0">
  <linkList><linkGrp targType="s"
fromDoc="Europarl/xml/eng/ep-00-01-17.xml"
  toDoc="Europarl/xml/fre/ep-00-01-17.xml">
      <link xtargets="1;1" />
      <link xtargets="2;2" />
      <link xtargets="3;3 4" />
```

Figure 1: Sentence alignments in LetsMT!

One of the main advantages of this approach is that alignment can be changed easily without the need of changing anything in the original corpus files. Various alignment versions can be stored and several languages can be linked together without repeating corpus data. Furthermore, corpus selection can be done using the same format. Several parallel corpora or only parts of certain corpora can be selected without the need of explicitly concatenating the corresponding corpus data. These selections can then be stored space-efficiently in the repository. They can be shared and revised easily. A simple procedure can then be used off-line to extract the actual data from the repository when training is initiated.

3. User-Tailored SMT Models

The largest benefit of the LetsMT! platform will be the support of user-specific SMT engines. Users of the platform will have full control over the selection of data resources which will be used for training a system. The potentials of such an approach can be seen in the test case described below.

We took data from the medical domain in order to show the impact of domain-specific data on SMT training. In particular we used the Swedish-English portion of the publicly available EMEA corpus which is part of OPUS (Tiedemann, 2009). This corpus covers a very specific domain including documents published by the European Medicines Agency. We extracted non-empty sentence alignments with a maximum of 80 tokens per sentence from the corpus in order to create appropriate training data for standard phrase-based SMT. Table 1 lists some statistics of the data.

	English	Swedish
sentences	898,359	898,359
tokens	11,567,182	10,967,600
unique sentence pairs		
sentences	298,974	298,974
tokens	4,961,225	4,747,807

Table 1: Training data extracted from EMEA

The EMEA corpus contains a lot of repetition as we can see from the numbers in table 1. The number of unique sentence pairs is much lower than the count for the original corpus. Naturally, we want to test the SMT model on unseen data only also to make a fair comparison to general-purpose machine translation. Therefore, we merged multiple occurrences of identical sentence pairs in order to create a set of unique sentence pairs and randomly selected 1000 of them for tuning and another 1000 for testing. The remaining sentence pairs are used for training. We trained standard phrase-based SMT models in both directions on that data using the target language side of the parallel training corpus for training the 5-gram language model. We basically used standard settings of the Moses system (Koehn et al., 2007) including lexicalized reordering and minimum error rate tuning.

For comparison we translated the same test set of 1000 example sentences using the current on-line system of Google Translate (date of the run: 28 August 2010) and measured lower-case BLEU scores for both systems. The results are shown in table 2.

	Google	Moses-EMEA
English-Swedish	50.23	59.29
Swedish-English	46.57	65.42

Table 2: Translation quality in terms of BLEU scores

The gain that we achieved by using in-domain training data is more impressive than we actually had expected. In the general case data of such a small size would not be sufficient for training appropriate SMT models. Not only the parallel data used for training the translation model is very little but especially the monolingual target language data used for the language model is much smaller than otherwise recommended. However, due to the domain specificity and especially the translation consistency in our data reasonable results can be achieved with this tiny amount of training data. Furthermore, we can see that general purpose translations do not reach the same quality even though they are trained on vastly larger amounts of data. It might even be possible that our training and test data is part of the collection used by Google as these documents are publicly available on the web. This, however, is beyond our control and we can only speculate about the resources used to train Google's translation engine.

4. References

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *ACL '07: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180, Morristown, NJ, USA.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia.

List of participants, SLTC 2010

Aarne Ranta, University of Gothenburg, aarne@chalmers.se
Aldin Draghoender, Stockholm University, aldind@hotmail.com
Anna Hjalmarsson, Tal, musik och hörsel, KTH, annah@speech.kth.se
Anna Prytz Lillkull, student KogVet, annpr075@student.liu.se
Annika Levin, Kronofogdemyndigheten, annika.levin@kronofogden.se
Annika Silvervarg, IDA, Linköpings universitet, annika.silvervarg@liu.se
Arne Jönsson, HCS/NLPLAB, arnjo@ida.liu.se
Ayanwale Olabisi, CAIRO UNIVERSITY, graceland58@yahoo.com
Barbro Ahlbom, Hjälpmedelsinstitutet, barbro.ahlbom@hi.se
Björn Granström, CTT/KTH, bjorn@speech.kth.se
Camilla Lindholm, Skatteverket, Camilla.lindholm@skatteverket.se
Caroline Willners, Oribi, caroline.willners@gmail.com
Charlotta Carlberg Eriksson, Skatteverket, Charlotta.carlberg.eriksson@skatteverket.se
Christian Smith, Linköpings Universitet, chrsm588@student.liu.se
Christian Wallin, Nota, cwa@nota.nu
Christina Tånnander, Talboks- och Punktskriftsbibliioteket (TPB), christina.tannander@tpb.se
Daniel Neiberg, CTT/TMH/KTH, neiberg@speech.kth.se
Daniel Scheidegger, MediaLT, daniel.scheidegger58@gmail.com
Dimitrios Kokkinakis, University of Gothenburg, dimitrios.kokkinakis@svenska.gu.se
Elena Yagunova, St.Petersburg State University, iagounova.elena@gmail.com
Elin Emsheimer, Kommunikationsmyndigheten PTS, elin.emsheimer@pts.se
Eva Orava, Forskningscentralen för de inhemska språken, eva.orava@focis.fi
Evamaria Nerell, Linköpings universitet, evamaria.nerell@gmail.com
Filip von Kartaschew, ReadSpeaker, filip.von.kartaschew@readspeaker.com
Fredrik Larsson, ReadSpeaker, fredrik.larsson@readspeaker.com
Gabriel Skantze, KTH Speech Music and Hearing, gabriel@speech.kth.se
Grégoire Détrez, Göteborg Universitet, gdetrez@crans.org
Gunhild Kvangarsnes, NLB, gunhild.kvangarsnes@nlb.no
Gunilla Nordling, CSN, gunilla.nordling@csn.se
Hans Engström, Eurocity, hasse@eurocitysoftware.com
Harald Hammarström, Radboud Universiteit Nijmegen and Max Planck Institute for Evolutionary Anthropology, h.hammarstrom@let.ru.nl
Henrik Danielsson, Linköping University, henrik.danielsson@liu.se
Henrik Haglund, CSN, henrik.haglund@csn.se
Henrik Nilsson, TNC, henrik.nilsson@tnc.se
Hercules Dalianis, DSV-Stockholm University, hercules@dsv.su.se
Hongzhan Hu, Linköping University, honhu753@student.liu.se
Håkan Jonsson, Voice Provider, hakan.jonsson@voiceprovider.com
Ioannis Iakovidis, Interverbum Technology, ioannis.iakovidis@interverbumtech.com
Jan Alexandersson, DFKI, Saarbrücken, janal@dfki.de

Janine Wicke , TPB, janine.wicke@tpb.se
Jens Erik Rasmussen, Mikro Værkstedet , jer@mikrov.dk
Joakim Gustafson, kth, jocke@speech.kth.se
Jody Foo, Linköping University, jody.foo@liu.se
Johanna Karlsson, Linköping University, johka299@student.liu.se
Johanne Ostad, The National Library of Norway (Språkbanken), Johanne.Ostad@nb.no
Jonas Rybing, Linköpings Universitet, jonry526@student.liu.se
Jörg Tiedemann, Uppsala University, jorg.tiedemann@lingfil.uu.se
Karin Friberg Heppin, Göteborgs universitet, karin.friberg@svenska.gu.se
Karin Husberg, Centrum för lättläst, karin.husberg@lattlast.se
Katarina Heimann Mühlenbock, Dept. of Swedish, University of Gothenburg, katarina.heimann.muhlenbock@gu.se
Keith Hall, Google Research, Zürich, kbhall@google.com
Kenneth Wilhelmsson, Högskolan i Borås, kenneth.wilhelmsson@hb.se
Kjetil Aasen, Språkrådet (Norge), kjetil.aasen@sprakradet.no
Kåre Sjölander, Readspeaker, kare.sjolander@readspeaker.com
Lars Ahrenberg, Linköpings universitet, lars.ahrenberg@liu.se
Lars Holmqvist, NåFram AB, larsa@nafram.se
Lena Stenberg, Mälardalens högskola, lena.stenberg@mdh.se
Lene Schmidt, Nordic Seminar on Language, lene.schmidt@rn.dk
Lidia Pivovarova, Saint-Petersburg State University, lidia.pivovarova@gmail.com
Lisa Ledin, Hjälpmedelsinstitutet, lisa.ledin@hi.se
Lise-Lott Andersson, Linköpings universitet, lise_lott.andersson@liu.se
Magne Lunde, MediaLT, magne@medialt.no
Magnus Merkel, Linköping University, magnus.merkel@liu.se
Maria Holmqvist, Linköpings universitet, maria.holmqvist@liu.se
Maria Toporowska Gronostaj, Univerity of Gothenburg, maria.gronostaj@svenska.gu.se
Marie Sandström, Linköpings Universitet, marie.sandstrom@liu.se
Marion Weller, IMS, Universität Stuttgart, wellermn@ims.uni
Maritha Angermund, Specialpedagogiska myndigheten, maritha.angermund@spsm.se
Martin Haulrich, Copenhagen Business School, mwh.isv@cbs.dk
Mats Wirén, Department of Linguistics, Stockholm University, mats.wiren@ling.su.se
Mattias Kanhov, Stockholm University, kanhov@gmail.com
Montserrat Arias , International Library - Stockholm, montserrat.arias@stockholm.se
Morten Tollefsen, MediaLT, morten@medialt.no
Ola Karlsson, Språkrådet, Ola.Karlsson@sprakradet.se
Olga Caprotti, University of Gothenburg, olga.caprotti@gu.se
Patrik Janesköld, Funka Nu AB, patrik.janeskold@funkanu.se
Per Langgård, Oqaasileriffik/ Sprogsekretariatet, Nuuk Grønland, per@oqaasileriffik.org
Per Starheim, MediaLT, per@medialt.no
Per-Anders Jande, Språkrådet, per.anders.jande@gmail.com
Peter Ljunglöf, DART och Språkbanken, GU, peter.ljunglof@gu.se
Pierre Nugues, Lunds universitet, LTH, Pierre.Nugues@cs.lth.se
Pär Gustavsson, Linköpings Universitet, pargu814@student.liu.se

Ramona Enache, University of Gothenburg, Chalmers University of Technology, ramona.enache@chalmers.se
Rickard Domeij, Språkrådet, Institutet för språk och folkminnen, rickard.domeij@sprakradet.se
Roar Nordby, MediaLT, roar@medialt.no
Robert Eklund, Voice Provider, robert.eklund@voiceprovider.com
Robert Krevers, University of Linköping, robkr997@student.liu.se
Robert Östling, Stockholms universitet, robert@ling.su.se
Robin Cooper,, Göteborgs universitet, cooper@ling.gu.se
Robin Keskisärkkä, Linköpings Universitet, robke281@student.liu.se
Sabine Kirchmeier-Andersen, Dansk Sprognaevn, sabine@dsn.dk
Sandra Derbring, None, sandra.derbring@gmail.com
Sara Stymne, Linköpings universitet, sara.stymne@liu.se
Sjur Nørstebø Moshagen, Sametinget i Norge, sjur.moshagen@samediggi.no
Sofia Bremin, student, sofia.bremin@gmail.com
Sofie Johansson Kokkinakis, Dept. of Swedish/Språkbanken/ISA, University of Gothenburg, sofie.johansson.kokkinakis@svenska.gu.se
Staffan Larsson, Göteborgs universitet, sl@ling.gu.se
Stefan Johansson, Funka Nu AB, stefan.johansson@funkanu.se
Stefan Pal, Mikro Værkstedet , stefan@mikrov.dk
Stina Ericsson, Talkamatic & Göteborgs universitet, stina@talkamatic.se
Sture Hägglund, Santa Anna IT Research Institute, stuha@ida.liu.se
Søren Axel Sørensen, Mikroverkstedet, sas@mikrov.dk
Torbjørg Breivik, Språkrådet, Norge, Torbjorg.Breivik@sprakradet.no
Torbjørn Nordgård, Lingit AS, torbjorn@lingit.no
Viggo Kann, KTH Teoretisk datalogi, viggo@nada.kth.se