NAACL HLT 2009

Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics

Proceedings of the Demonstration Session

Fred Popowich and Michael Johnston
Demo Chairs

June 1-3, 2009
Boulder, Colorado

Production and Manufacturing by
Omnipress Inc.
2600 Anderson Street
Madison, WI 53707
USA

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-932432-42-8

Organizers:

Michael Johnston, AT&T Labs Research
Fred Popowich, Simon Fraser University

Table of Contents

Conference Program

Monday, June 1, 2009

6:30pm–9:30pm Demonstration Session

Cross-document Temporal and Spatial Person Tracking System Demonstration
Heng Ji and Zheng Chen

Building Conversational Agents with Basilica
Rohit Kumar, Carolyn P. Rosé and Michael J. Witbrock

STAT: Speech Transcription Analysis Tool
Stephen A. Kunath and Steven H. Weinberger

Morpho Challenge - Evaluation of algorithms for unsupervised learning of morphology in various tasks and languages
Mikko Kurimo, Sami Virpioja, Ville Turunen and Teemu Hirsimäki

WordNet::SenseRelate::AllWords - A Broad Coverage Word Sense Tagger that Maximizes Semantic Relatedness
Ted Pedersen and Varada Kolhatkar

Cross-document Temporal and Spatial Person Tracking System Demonstration

Heng Ji
Queens College and the Graduate Center

Zheng Chen
The Graduate Center

The City University of New York
New York, NY, 11367
hengji@cs.qc.cuny.edu
zchen1@gc.cuny.edu

Abstract

Traditional Information Extraction (IE) systems identify many *unconnected* facts. The objective of this paper is to define a new cross-document information extraction task and demonstrate a system which can extract, rank and track events in two dimensions: temporal and spatial. The system can automatically label the person entities involved in significant events as '*centroid arguments*', and then present the events involving the same centroid on a time line and on a geographical map.

1 Introduction

Information Extraction (IE) systems can identify 'facts' (entities, relations and events) of particular types within individual documents, and so can unleash the knowledge embedded in texts for many domains, such as military monitoring, daily news, financial analysis and biomedical reports. However, most current IE systems focus on processing single documents and, except for coreference resolution, operate a sentence at a time. The result are large databases containing many *unconnected, unranked, redundant* (and some *erroneous*) facts.

McNamara (2001) proved that a high-coherence text has fewer conceptual gaps and thus requires fewer inferences and less prior knowledge, rendering the text easier to understand. In our task text coherence is the extent to which the relationships between events in a text can be made explicit. We noted that linking all events in temporal and spatial directions for the entire corpus was not feasible because of the large number of event arguments. Grosz et al. (1995) claimed that certain entities are more central than others and that this property imposed constraints on discourse coherence. Therefore we have developed a system which can extract globally salient and novel arguments as *centroid arguments*, and link all events involving each centroid argument on a time line and on a geographical map.

Beyond extracting isolated facts from individual sentences, we provide coherent event chains so that the users can save time in connecting relevant events and conducting reasoning, such as tracking a person's movement activities and an organization's personnel changes. This will provide a richer set of views than is possible with document clustering for summarization or with topic tracking. In addition, such cross-document extraction results are indexed and allow a fast entity searching mechanism. Beyond traditional search, the system can correlate and organize information across different time series by temporal tracking, and deliver to users in different geographies by spatial tracking.

The rest of this paper is structured as follows. Section 2 presents the overall system architecture including the baseline system and the detailed approaches to extract event chains. Section 3 then presents the experimental results compared to traditional IE. Section 4 demonstrates the system output. Section 5 compares our approach with related work and Section 6 then concludes the paper and sketches our future work.

2 System Overview

In this section we will present the overall procedure of our system.

Proceedings of NAACL HLT 2009: Demonstrations, pages 1–4,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

2.1 Within-document IE

We first apply a state-of-the-art English IE system (Ji and Grishman, 2008) to extract events from each single document. The IE system includes entity extraction, time expression extraction and normalization, relation extraction and event extraction. Entities include persons, locations, organizations, facilities, vehicles and weapons; Events include the 33 distinct event types defined in Automatic Content Extraction (ACE05)[1].

The event extraction system combines pattern matching with statistical models. For every event instance in the ACE training corpus, patterns are constructed based on the sequences of constituent heads separating the trigger and arguments. In addition, a set of Maximum Entropy classifiers are trained: to distinguish events from non-events; to classify events by type and subtype; to distinguish arguments from non-arguments; to classify arguments by argument role; and given a trigger, an event type, and a set of arguments, to determine whether there is a reportable event mention. In addition, the global evidence from related documents is combined with local decisions to conduct cross-document inference for improving the extraction performance as described in (Ji and Grishman, 2008).

2.2 Centroid Argument Detection

After we harvest a large repository of events we can label those important person entities which are involved frequently in events as '*centroid arguments*'. Not only are such arguments central to the information in a collection (high-frequency), they also should have higher accuracy (high-confidence). In this project we exploit global confidence metrics to reach both of these two goals.

For an event mention, the within-document event classifiers produce the following local confidences values:

- *LConf(trigger,etype)*: The probability of a string *trigger* indicating an event mention with type *etype*.
- *LConf(arg, etype)*: The probability that a mention *arg* is an argument of some particular event type *etype*.
- *LConf(arg, etype, role)*: If *arg* is an argument with event type *etype*, the probability of *arg* having some particular *role*.

We use the INDRI information retrieval system (Strohman et al., 2005) to obtain the top N related documents for each test document to form a *topically-related cluster*. The intuition is that if an argument appears frequently as well as with high extraction confidence in a cluster, it is more salient. For each argument *arg* we also added other person names coreferential with or bearing some ACE relation to the argument as *argset*.

In addition we developed a cross-document person name disambiguation component based on heuristic rules to resolve ambiguities among centroid arguments. Then we define the following global metric weighted with the local confidence values to measure *salience*, and generate the top-ranked entities as *centroid arguments*.

- *Global-Confidence(arg):* The frequency of *argset* appearing as an event argument in a cluster, weighted by local confidence values: *LConf(trigger,etype)*LConf(arg, etype)* LConf(arg, etype, role).*

2.3 Cross-document Event Aggregation and Global Time Discovery

If two events involve the same centroid argument, we order them along a time line according to their time arguments and group them into specific geographical locations based on their place arguments. When ordering a pair of entity arguments, we replace pronouns with their coreferential names or nominals, and replace nominals with their coreferential names, if applicable. If the normalized dates are the same for two events, we further compare them based on their time roles (e.g. 'time-end' should be ordered after 'time-beginning').

We start from aggregating events by merging coreferential event mentions using the within-document coreference resolution component in the IE system. However, the degree of similarity among events contained in a group of topically-related documents is much higher than within a document, as each document is apt to describe the main point as well as necessary shared background.

[1] http://www.nist.gov/speech/tests/ace/

Relation	$Event_i$ Arguments	$Event_j$ Arguments	Centroid	Event Type	Event Time
Coreference	Entity[Ariel Sharon] Place [Jerusalem]	Entity[Sharon] Place[Jerusalem]	Powell	Contact-Meet	2003-06-20
Subset	Entity[Bush]	Entity[Bush] Place[Camp David]	Blair	Contact-Meet	2003-03-27
Subsumption	Destination[Mideast]	Destination[Egypt]	Bush	Movement-Transport	2003-06-02
Complement	Sentence [nine-year jail] Crime[corruption]	Adjudicator[court] Place[Malaysia] Sentence [nine-year prison]	Anwar Ibrahim	Justice-Sentence	2003-04-18

Table 1. Cross-document Event Aggregation Examples

Therefore in order to maximize *diversity*, we merge any pair of events that have the same event type and involve the same centroid argument, via one of the operations in Table 1.

3 Experimental Results

We used 10 newswire texts from ACE 2005 training corpora as our test. For each test text we retrieved 25 related texts from English Topic Detection and Tracking (TDT-5)[2] corpus which in total consists of 278,108 texts. The IE system extracted 179 event mentions including 140 Name arguments. We define an argument is correctly extracted if its event type, offsets, and role match any of the reference argument mentions.

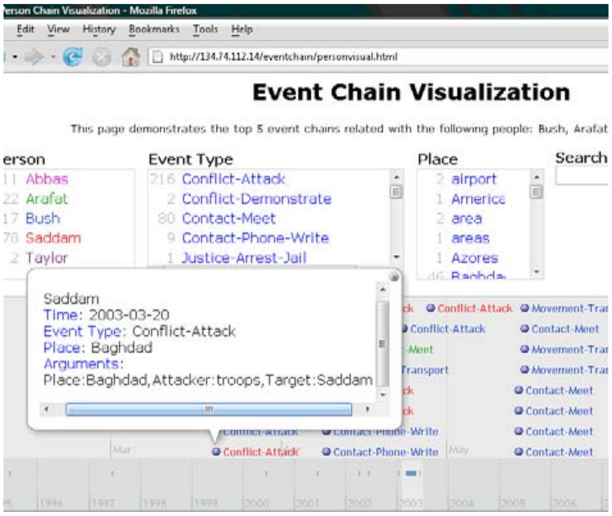
We found that after ranking with the global confidence metrics, the top-ranked event arguments are substantially more accurate than the arguments as a whole: the overall accuracy without ranking is about 53%; but after ranking the top 85 arguments (61% of total) get accuracy above 70% and the top 116 arguments (83% of total) are above 60% accuracy. It suggests that aggregating and ranking events according to global evidence can enable users to access salient and accurate information rapidly.

4 Demonstration

In this section we will demonstrate the results on all the documents in the English TDT5 corpus. In total 7962 person entities are identified as centroid arguments. The offline processing takes about three hours on a single PC. The real time browsing only takes one second in a standard web browser.

Figure 1 and Figure 2 present the temporal and spatial event chains involving the top 5 centroid arguments: “Bush”, “Arafat”, “Taylor”, “Saddam” and “Abbas”. The events involving each centroid are ordered on a time line (Figure 1) and associated with their corresponding geographical codes in a map (Figure 2).

The users can drag the timeline and map to browse the events. In addition, the aggregated event arguments are indexed and allow fast centroid searching. Each argument is also labeled by its global confidence, language sources, and linked to its context sentences and other event chains it is involved. We omit these details in these screenshots.



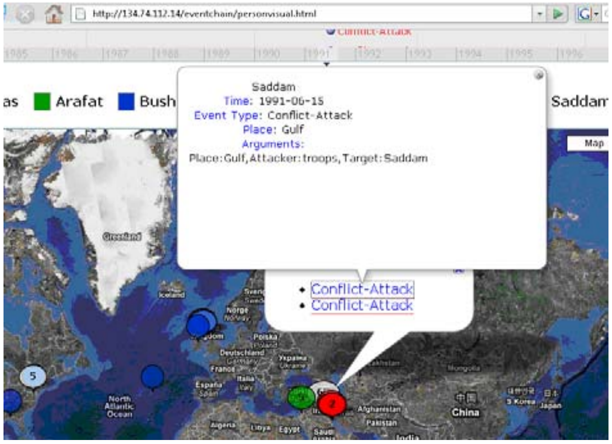
Event Chain Visualization

This page demonstrates the top 5 event chains related with the following people: Bush, Arafat

Saddam
Time: 2003-03-20
Event Type: Conflict-Attack
Place: Baghdad
Arguments:
Place:Baghdad,Attacker:troops,Target:Saddam

Figure 1. Temporal Person Tracking

[2] http://projects.ldc.upenn.edu/TDT5/



Saddam
Time: 1991-06-15
Event Type: Conflict-Attack
Place: Gulf
Arguments:
Place:Gulf,Attacker:troops,Target:Saddam

Figure 2. Spatial Person Tracking

5 Related Work

Recently there has been heightened interest in discovering temporal event chains. For example, Bethard and Martin (2008) applied supervised learning to classify temporal and causal relations simultaneously. Chambers and Jurafsky (2008) extracted narrative event chains based on common protagonists. In this paper we import these ideas into IE while take into account some major differences. Following the original idea of centering (Grosz et al., 1995) and the approach of centering events involving protagonists (Chambers and Jurafsky, 2008), we introduce a new concept of 'centroid arguments' to represent those entities which are involved in all kinds of salient events frequently. We operate cross-document instead of within-document, which requires us to resolve more conflicts and ambiguities. In addition, we study the temporal and spatial linking task on top of IE results. In this way we extend the representation of each node in the chains to a structured aggregated event including fine-grained information such as event types, arguments and their roles.

6 Conclusion and Future Work

In this paper we described several new modes for browsing and searching a large collection of news articles, and demonstrated a system implementing these modes. We introduced ranking methods into IE, so that the extracted events are connected into temporal and spatial chains and presented to the user in an order of *salience*. We believe these new forms of presentation are likely to be highly beneficial, especially to users whose native language is not English, by distilling the information landscape contained in the large collection of daily news articles – making more information sources accessible and useful to them.

On the other hand, for the users searching news about particular person entities, our system can suggest a list of centroid event arguments as key words, and provide a brief story by presenting all connected events. We believe this will significantly speed up text comprehension. In this paper we only demonstrated the results for person entities, but this system can be naturally extended to other entity types, such as company names to track their start/end/acquire/merge activities. In addition, we plan to automatically adjust cross-document event aggregation operations according to specific compression ratios provided by the users.

Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency under Contract No. HR0011-06-C-0023 via 27-001022, and the CUNY Research Enhancement Program and GRTI Program.

References

Steven Bethard and James H. Martin. 2008. Learning semantic links from a corpus of parallel temporal and causal relations. *Proc. ACL-HLT 2008*.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains. *Proc. ACL 2008*.

Barbara Grosz, Aravind Joshi, and Scott Weinstein. 1995. Centering: A Framework for Modelling the Local Coherence of Discourse. *Computational Linguistics*, 2(21), 1995.

Heng Ji and Ralph Grishman. 2008. Refining Event Extraction Through Unsupervised Cross-document Inference. *Proc. ACL 2008*.

Danielle S McNamara. 2001. Reading both High-coherence and Low-coherence Texts: Effects of Text Sequence and Prior Knowledge. *Canadian Journal of Experimental Psychology*.

Trevor Strohman, Donald Metzler, Howard Turtle and W. Bruce Croft. 2005. Indri: A Language-model based Search Engine for Complex Queries (extended version). *Technical Report IR-407, CIIR*, Umass Amherst, US.

Building Conversational Agents with Basilica

Rohit Kumar **Carolyn P. Rosé**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
rohitk@cs.cmu.edu cprose@cs.cmu.edu

Abstract

Basilica is an event-driven software architecture for creating conversational agents as a collection of reusable components. Software engineers and computer scientists can use this general architecture to create increasingly sophisticated conversational agents. We have developed agents based on Basilica that have been used in various application scenarios and foresee that agents build on Basilica can cater to a wider variety of interactive situations as we continue to add functionality to our architecture.

1 Introduction

Conversational Interfaces apply the metaphor of agent to an interface which allows the user to conversationally interact with the machine using natural language through speech or text. The current state of the art in the area of conversational interfaces is largely dominated by spoken dialog systems (SDS). These SDS are most often used for the purpose of accessing information from a database over the telephone. Other common applications of conversational agents include computer aided instruction (CAI) and human-robot interaction (HRI).

Conversational Agents in most of today's SDS, CAI and HRI are designed to work within the scope of specific task domains which allows the scientists and engineers working on such systems to ensure satisfactory and relevant interaction with the user most of the time. Within the task domain, such agents can display intelligent interactive behavior like helping the user use the interface, asking remedial questions (Bohus and Rudnicky, 2005), shaping the user behavior (Tomko and Rosenfeld, 2004) by using alternative phrasing of utterances, responding to user affect (D'Mello et al., 2008) through text, voice and gesture, engaging the user through the display of presence via backchannels (Ward, 1996) and embodiment (Cassell et al., 1999).

As more and more of these intelligent interactive agents get built for many task domains (Raux et al., 2005; Bohus et al., 2007; Gockley et al., 2005; Amtrak Julie; …) that surround our everyday life, we observe a gradual transition in the use of the conversational agent technology to be a form of situated interaction. One of the characteristic requirements of this transition towards ubiquity of such interactive agents is the capability to sense and trigger behavior in a context sensitive way.

In most conversational interfaces today, the only trigger used by the agents is that of initiation of conversation usually by sensing user presence through a telephone call, proximity detection or user login into a virtual environment. The initiation event is followed by a scripted task-oriented conversation with the agent. These scripts could be fairly complex depending on the representational formalism underlying the script. Most of the common software architectures/platforms used to create conversational agents like TellMe Studio, Voxeo Prophecy, Olympus (Bohus et al., 2007), DIPPER (Bos and Oka, 2003), etc. use one or more of these presence sensing techniques and one of the many existing scripting languages including VoiceXML, SALT, TuTalk (Jordan et al., 2007) and Ravenclaw (Bohus and Rudnicky, 2003) task specification language among others.

However, in our recent work on building conversational agents situated in collaborative learning

Proceedings of NAACL HLT 2009: Demonstrations, pages 5–8,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

environments, we have discovered the need for a software architecture for creating agents that persist in an interactive environment in which human users interact with these agents as well as with each other. In this situation, the agents need to be able to sense many kinds of triggers at many points of time and choose to respond to some of those triggers through a variety of modalities including conversation. This observation was the motivation for creating Basilica which is our architecture for building conversational agents. In section 2, we talk more about the intricacies of Basilica and agents built on this architecture. Section 3 describes some of application scenarios in which we are using Conversational Agents based on Basilica.

2 Basilica Architecture

In order to meet the need for an architecture that enables development of Conversational Agents as a collection of behavioral components that can sense triggers and respond to those appropriately, we created the Basilica architecture.

In this architecture, we model sensing and responding as two types of components that make up conversational agents. The sensing components referred to as *Filters* observe stimuli from various kinds of input sources and other components. They can also generate stimuli for other components. On the other hand, *Actor* components generate responsive behavior that may be observed the user(s) and other components. Basilica provides the software elements required to tie Filters and Actors together through *Connections* that carry *Events* over them. We think that many of the state of the art intelligent behaviors listed in section 1 can be implemented as dyads of filter and actor components.

The minimal set of behavioral component classes listed above can easily be extended. For example, certain agent designs may need memory components and coordination components which bridge across multiple actors or filters that do not necessarily share events with each others. Timer components may be used to generate regulated stimuli. Besides belonging to one of these classes of components, certain components may act as wrappers to external systems. For example, we use wrapper components to integrate TuTalk dialog management system (Jordan et al., 2007) for some of the instructive behavior exhibited by our agents. Also, certain components act as wrappers to the environment in which the agent is present. These wrappers help in easily integrating the same agent with multiple environments without having to change any underlying components except the wrappers to the environment.

We believe that fairly intelligent conversational agents can be built for situated interaction applications by incrementally building a large number of behavioral components. Each of these components represent a decomposition of the agent's perceptive and cognitive capabilities. Among the agents we have built using Basilica, we observe that some of these capabilities are common across agents. Hence the corresponding behavioral components get re-used in many cases. Some instances of component re-use are mentioned in Section 3.

Note that recently there has been other work on modeling conversational agents as a decomposition of components. Jaspis (Turunen and Hakulinen, 2003) models the agent as a collection of *managers*, *agents* and *evaluators* which synchronize with each other through *transactions*. RIME (Nakano et al., 2008) distributes cognitive capabilities across a collection of *experts* of two types. However, *evaluators* and *agents* are configured as a pile of components whereas our filters and actors are configured as a network. Hence, designing conversational agents with Basilica gives the flexibility to change the network topology. Also, while Jaspis agents are stateless, actors in our architecture need not be stateless. In other work on event-based multi-layered architectures (Raux and Eskenazi, 2007), events are used for communication between layers as a mean to provide higher reactive compared to pipeline architectures. While we share this motivation, definition of events is extended here as events are used for all kinds of communication, coordination and control in Basilica.

3 Current Application Scenarios

In 2008, we built three conversational agents to support learners in collaborative learning environments. Also, we are currently using Basilica to develop a cross-lingual assistive agent to support non-Spanish speaking 911 dispatchers in the southern states of the US. In this section, we will discuss these four conversational agents briefly.

CycleTalk is an intelligent tutoring system that helps college sophomores studying Thermodynamics learn about principles of designing Steam cycles. In our recent experiments, we have studied the effectiveness of conversational agents in this intelligent tutoring system (Kumar et al., 2007; Chaudhuri et al., 2008). Student use the system both individually and in pairs. The conversational agent monitors student interaction in a chat room as the students work on solving a design problem. The tutor provides the students with hints to help touch upon all the underlying concepts while the students work on the design exercise. Also the agent brings up reflective dialogs when it detects a relevant topic in the students conversation. One of the problems we observed over the years with the use of instructional dialogs in collaborative environments is that the students tend to ignore the tutoring agent if it interrupts the students when they are talking to each other. Basilica helped us in resolving this problem by implementing a component that tells that student that help is available on the topic they are talking about and they can ask for the dialog support when they are ready. Basilica gives the flexibility to change the intervention strategy used by the agent when it is speaking with more than one student.

In another version of this system, the tutoring agent prompted the students with some motivational prompts occasionally as we observed that many of the students found the design exercise very demanding to complete in the time permitted for this lab exercise. We found that the use of motivational prompts improved the student's attitude towards the automated agent.

We developed another agent to help college level mathematics students working on problem solving. This agent operates in a collaborative environment which includes a whiteboard. As in the case with the CycleTalk agent, the agent used here also helps the students with hints and dialogs. The component required for those behaviors were re-used as-is with modifications only their configuration files. Besides these behaviors, the agent coordinates the problem solving sessions for the team by presenting the team with problems as images placed on the whiteboard and helping the students stay on track by answering questions about the amount of time left in the problem solving session.

Recently, we modified the environment wrapper components of our CycleTalk agent and integrated them with a SecondLife application (Weusijana et al., 2008). This integration helps developers of conversational agents create interactive agents in the SecondLife virtual environment.

Finally, in a currently ongoing project, we are building an agent that would interpret Spanish utterances from a distressed 9-1-1 caller and work with a human dispatcher who does not know Spanish to attend to the call. We model the agent in this scenario after a human translator who does not just translate the caller's input to English and vice versa. Instead the translator partners with the dispatcher to provide service to the caller. Partnering conversational agents with a human user to help another human user in a different role is a novel application of interactive agents.

4 Building Agents using Basilica

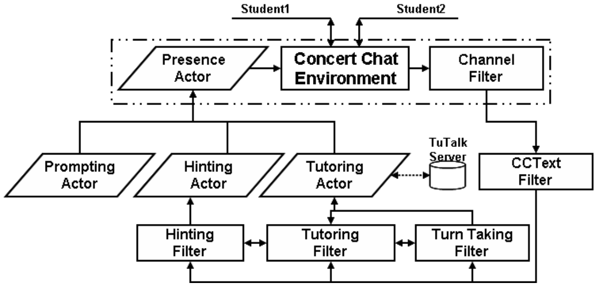


Figure 1. Components of the CycleTalk Agent

Building conversational agents using Basilica involves the process of representing the desired agent as a decomposition of components. Figure 1 above shows the components that make up the CycleTalk conversational agent we mentioned in Section 3. The rectangles represent Filters and the parallelograms represent Actors. Connections are shown as solid lines. In a detailed design, these lines are annotated with the events they carry.

Once an agent is designed, the agents and filters required for the implementation of the agent can be either re-used from the pre-existing components of Basilica or implemented as Java objects that extend the corresponding component class. Often the programming task is limited to implementing handlers and generators for the events received and sent out by the component. Theoretically, the validity of a component can be verified if it can handle and generate all the events as specified in the design diagram.

As we continue to develop more conversational agents on this architecture, we intend to create development tools which would easily translate a design like Figure 1 to the implementation and facilitate validation and debugging of the agent.

5 Demonstration Outline

The demonstration of our architecture will give the audience an opportunity to interact with the agents we have described in section 3 and discuss how we can design such agents using Basilica. We will have a poster to aid the discussion along with ability to probe into the code underlying the design of these agents. Attendees will be able to understand the process involved in building agents with Basilica and assess the effort required. Additionally, if we have any specialized development tools to automatically map agent design as described in Section 4 to Java code, we will demonstrate those tools. Up to date information about Basilica can be found at http://basilica.rohitkumar.net/wiki/

Acknowledgements

This work is supported by NSF REESE/REC grant number 0723580.

References

Dan Bohus and Alex Rudnicky, 2005. *Error Handling in the RavenClaw dialog management architecture*, HLT-EMNLP-2005, Vancouver

Stefanie Tomko and Roni Rosenfeld, 2004. *Shaping Spoken Input in User-Initiative Systems*. Interspeech 2004, Jeju, Korea

Antoine Raux, Brian Langner, Dan Bohus, Alan Black, and Maxine Eskenazi, 2005. *Let's Go Public! Taking a Spoken Dialog System to the Real World*, Interspeech 2005, Lisbon, Portugal

Dan Bohus, Sergio Grau, David Huggins-Daines, Venkatesh Keri, Gopala Krishna A., Rohit Kumar, Antoine Raux, and Stefanie Tomko, 2007. *Conquest - an Open-Source Dialog System for Conferences*, HLT-NAACL 2007, Rochester, NY

Amtrack Julie, http://www.networkworld.com/news/ 2003/0619julie.html

Justin Cassell, Timothy Bickmore, Billinghurst, M., Campbell, L., Chang, K., Vilhjálmsson, H. and Yan, H., 1999. *Embodiment in Conversational Interfaces: Rea*, CHI'99, Pittsburgh, PA

Nigel Ward, 1996. *Using Prosodic Clues to decide when to produce Back-channel Utterances,* ICSLP 96

Sidney D' Mello, Tanner Jackson, Scotty Craig, Brent Morgan, Patrick Chipman, Holly White, Natalie Person, Barry Kort, Rana el Kaliouby, Rosalid W. Picard and Arthur Graesser, 2008, *AutoTutor Detects and Responds to Learners Affective and Cognitive States*, Workshop on Emotional and Cognitive Issues, ITS 2008, Montreal

Rachel Gockley, Allison Bruce, Jodi Forlizzi, Marek Michalowski, Anne Mundell, Stephanie Rosenthal, Brennan Sellner, Reid Simmons, Kevin Snipes, Alan C. Schultz and Jue Wang, 2005. *Designing Robots for Long-Term Social Interaction*, IROS 2005

Dan Bohus, Antoine Raux, Thomas Harris, Maxine Eskenazi and Alex Rudnicky, 2007. *Olympus: an open-source framework for conversational spoken language interface research* HLT-NAACL 2007 Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technology, Rochester, NY

Johan Bos and Tetsushi Oka, 2003. *Building Spoken Dialogue Systems for Believable Characters*, 7th workshop on the semantics & pragmatics of dialogue

TellMe, https://studio.tellme.com/

Voxeo Prophecy, http://www.voxeo.com/products/

Pamela Jordan, Brian Hall, Michael Ringenberg, Yue Cui, Carolyn P. Rosé, 2007. *Tools for Authoring a Dialogue Agent that Participates in Learning Studies*, AIED 2007

Dan Bohus and Alex Rudnicky, 2003. *RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda*, Eurospeech 2003, Geneva, Switzerland

Markku Turunen, Jaakko Hakulinen, 2003. *Jaspis - An Architecture for Supporting Distributed Spoken Dialogues*, Eurospeech' 2003, Geneva, Switzerland

Mikio Nakano, Kotaro Funakoshi, Yuji Hasegawa, Hiroshi Tsujino, 2008. *A Framework for Building Conversational Agents Based on a Multi-Expert Model*, 9th SigDial Workshop on Discourse and Dialog, Columbus, Ohio

Antoine Raux and Maxine Eskenazi, 2007. *A Multi-Layer Architecture for Semi-Synchronous Event-Driven Dialogue Management*, ASRU 2007, Kyoto

Rohit Kumar, Carolyn Rose, Mahesh Joshi, Yi-Chia Wang, Yue Cui, Allen Robinson, *Tutorial Dialogue as Adaptive Collaborative Learning Support*, 13th AIED 2007, Los Angeles, California

Sourish Chaudhuri, Rohit Kumar, Carolyn P. Rose, 2008. *It's not easy being green - Supporting Collaborative Green Design Learning*, ITS 2008, Montreal

Baba Kofi A. Weusijana, Rohit Kumar, Carolyn P. Rose, 2008. *MultiTalker: Building Conversational Agents in Second Life using Basilica*, Second Life Education Community Convention, Purple Strand: Educational Tools and Products, 2008, Tampa, FL

STAT: Speech Transcription Analysis Tool

Stephen A. Kunath
Program in Linguistics
3e4 George Mason University
Fairfax, VA 22030
skunath@gmu.edu

Steven H. Weinberger
Program in Linguistics
3e4 George Mason University
Fairfax, VA 22030
weinberg@gmu.edu

Abstract

The Speech Transcription Analysis Tool (STAT) is an open source tool for aligning and comparing two phonetically transcribed texts of human speech. The output analysis is a parameterized set of phonological differences. These differences are based upon a selectable set of binary phonetic features such as [voice], [continuant], [high], etc. STAT was initially designed to provide sets of phonological speech patterns in the comparisons of various English accents found in the Speech Accent Archive http://accent.gmu.edu, but its scope and utility expand to matters of language assessment, phonetic training, forensic linguistics, and speech recognition.

1 Introduction

The theoretical and practical value of studying human accented speech is of interest to language teachers, linguists, and computational linguists. It is also part of the research program behind the Speech Accent Archive (http://accent.gmu.edu) housed at George Mason University. The Archive is a growing database of English speech varieties that contains more than 1,100 samples of native and non-native speakers reading from the same English paragraph. The non-native speakers of English come from more than 250 language backgrounds and include a variety of different levels of English speech abilities. The native samples demonstrate the various dialects of English speech from around the world. All samples include phonetic transcriptions, phonological generalizations, demographic and geographic information. For comparison purposes, the Archive also includes phonetic sound inventories from more than 200 world languages so that researchers can perform various contrastive analyses and accented speech studies.

No matter how subtle an accent is, human listeners can immediately and automatically notice that speakers are different. For example, Chinese speakers of English sound different from French speakers of English. The Speech Accent Archive stores and presents data that specifies and codifies these speech differences at the phonetic segment level. Trained human linguists compare a standard speech sample with phonetically transcribed speech samples from each (non-standard or non-native) speaker and distill from this analysis a set of phonological speech patterns (PSPs) for each speaker. Essentially, the task is to discover the precise factors or features responsible for humans to categorize say, a Vietnamese speaker of English differently from a so-called standard English speaker. While such analyses are theoretically and practically valuable, the process of comparing two phonetically transcribed speech samples requires explicit training, is time-consuming, and is difficult to update.

2 Phonological Speech Patterns

As an example of how we manually derive the PSPs for a non-native English speaker, we begin by comparing the narrow phonetic transcription of a “standard” North American English sample (1), with a representative non-native speaker of English (here a Vietnamese speaker (2)):

(1) [p^{h}l̥iiːz k^{h}ɑl^{ɣ} stɛlə æskɚ ɾə bɹ̃ĩŋ ðiiːz θĩŋz wɪθɚ fɹʌ̃m ðə stɔɹ sɪks spũunz əv fɹɛʃ snoʊ p^{h}iiːz faɪːv θɪk sl̥æːbz əv bluː ʧiiːz æn meɪbi ə snæk˺ fɚ hɚ bɹʌðɚ bɑːb wii ɑl^{ɣ}so niĩːɾə smɑl^{ɣ} p^{h}l̥æstɪk˺ sneɪk æ̃nə bɪːg t^{h}ɔɪ fɹ̥ɑːg fɚ ðə k^{h}ɪːdz

Proceedings of NAACL HLT 2009: Demonstrations, pages 9–12,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

ʃii kə̃n skʷuup˺ ðii:z θɪŋ̃z ɪ̃ntə θɹ̥ii ɹɛːd˺ bæːgz æ̃ːn wii wɪlˠ goʊ miit hɚ wɛ̃nzdeɪ æt˺ ðə tʰɹ̥eĩn steɪʃə̃n]

(2) [pli kolˠ stɛlɔ as xɜ t̠ʊ bɹɪŋ ði θɪŋgs wɪd̪ xɜː fɹɔm ə s̪tɔː sɪxs spuːn ɔf fɹɛʃ noʊ piːz faiθ t̪ɪk ə̆slæp˺ ɔ βlu çiːs ẽn meɪbi ɛ snæk˺ fɔ xɜː bɹʌðə bɔʔ wi ɔlˠsɔ niːt ʔʌ psmɔːlˠ plæstɪk snex ɛnʌ bix tɔɪ fɹɔx fɔ ðə kiːs ʃi kʲẽːn skuʔ lɪ θʰɪŋgs ɪntʊ tɹiː ɹɛd̥ bæɣz̥ ɛn wĭ wil go mit˺ xɜ wɛnz̥deɪ a ðəs tɹeĩn steɪʃɪn]

Each of these phonetic transcriptions are constructed by 3 to 4 trained linguists, and disagreements are settled by consensus. As is the case with all such transcriptions, they remain works in progress. Two of these trained linguists do a pencil and paper word-by-word comparison of the two transcriptions in (1) and (2). Their analysis of the data may find the following PSPs listed in (3):

(3) (a) final obstruent devoicing ([çiːs])

(b) non aspiration ([piːz])

(c) final consonant deletion ([pli])

(d) vowel epenthesis ([ə̆slæp˺])

(e) substitution of [x] for velars and glottals ([bix])

This is just a partial list. Some speakers may have more, and some speakers may have less. But the essential claim here is that each speaker's English accent is the sum of their PSPs.

There are certain problems associated with this manual process. Foremost among them is the cost and time to train linguists to perform uniform PSP analyses. Analysts must know what to look for—they must decide what is important and what should be ignored. This brings us to the second drawback of manual analysis: the lack of a quick and parameterized method of comparison.

If researchers need to test hypotheses about additional but uncatalogued PSPs, or if they need to simply search for a defined subset of PSPs, additional manual analyses are necessary. A third problem appears in the proper selection of one arbitrary standard "base" sample for the comparisons. At times researchers may want to compare non-natives with American English native samples, and at other times they may need to compare non-natives with British, or other varieties of native English. This requires multiple manual comparisons, and they take human time and energy. Finally, as mentioned above, narrow phonetic transcriptions may need to be modified as collaborators join the analysis. But when these are changed, they necessitate concomitant change in the register of PSPs.

Automating PSP generation not only solves these problems, but also opens up new research possibilities.

3 An Automated System: Research Potentials

We have developed a computational tool that will automatically compare two phonetically-transcribed speech samples and generate a set of PSPs describing the speech differences. Automating the comparison process will be of great use to the archive and to any speech scientist who transcribes and analyzes spoken language. It will allow fast and pointed comparisons of any two phonetically transcribed speech samples. Instead of simply comparing a "standard" North American native speaker and a non-native speaker, it will be quite simple to perform many accent comparisons, including those between a native British English speaker and a non-native speaker. It will also be possible to quickly and easily derive a composite result. That is, after a number of analyses, we can determine what a typical Russian speaker of English will do with his vowels and consonants. This promises to be a great empirical improvement over the pronouncements that are currently offered in the appendices of various ESL teacher-training textbooks.

For the analysis of individual speakers, this tool has direct use in matters of linguistic assessment. It will be useful in the fields of ESL pronunciation assessment (Anderson-Hsieh, Johnson, and Kohler, 1992). These kinds of assessments will naturally lead to a theory of *weighted* PSPs.

The tool also serves as a fast and systematic method of checking human transcription accuracy and thereby facilitates better methods of phonetic transcription (Cucchiarini, 1996; Shriberg, Hinke, & Trost-Steffen, 1987).

Finally, the tool can provide a needed human factor diagnostic to guide research in spectrographic speech analysis. And because speech recognition and speaker identification programs must ultimately deal with different accented speech, the results from the STAT analyses will contribute to this work (Bartkova & Jouvet, 2007; Deshpande, Chikkerur, & Govindaraju, 2005).

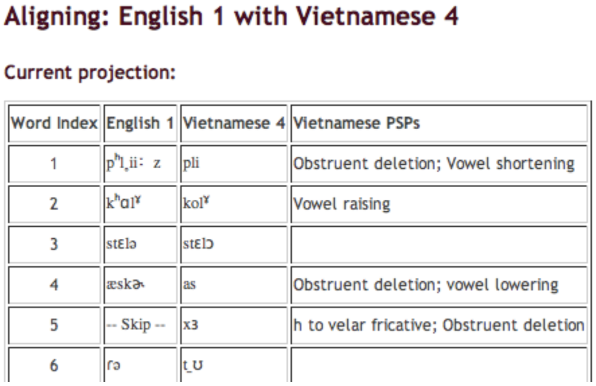
4 System Overview

Linguists who transcribe speech into a phonetic representation may use a tool such as PRAAT, to play the audio source file and a text editor to input the transcription. The result is normally a Unicode text file that has an IPA transcription of the audio file. STAT provides linguists with an easy way to play back an audio source file and share it with other linguists. A key feature that STAT provides in addition to transcription tools is a mechanism to manage a corpus of phonetic transcriptions. Once a corpus of phonetic transcriptions is created, linguists can use STAT's phonological speech pattern analysis tools to describe differences between different speakers' accents.

The STAT system incorporates several distinct components. Users interact with the system primarily via a web interface. All user interfaces are implemented with Ruby on Rails and various JavaScript libraries. Backend processes and algorithms are implemented in Java. An open source web application bundle including the front-end web interfaces and backend libraries will be made available as an open source library suitable for use in other applications in the future. We believe that the transcription alignment and speech pattern analysis components of STAT make it a unique tool for linguists studying speech processes.

4.1 Language Management

The language management component of STAT provides basic transcribed audio corpus management. This module allows a user to define a new speaker source language, e.g. Japanese, and specify attributes of the language, e.g. a phonetic inventory. All transcriptions are then associated with a speaker source language. STAT offers robust search capabilities that allow a linguist to search by things such as speaker demographics, phonetic inventories, phonological speech processes, and speech quality assessments.



Aligning: English 1 with Vietnamese 4

Current projection:

Word Index	English 1	Vietnamese 4	Vietnamese PSPs
1	pʰl̥iiː z	pli	Obstruent deletion; Vowel shortening
2	kʰɑɪˠ	kolˠ	Vowel raising
3	stɛlə	stɛlɔ	
4	æskɚ	as	Obstruent deletion; vowel lowering
5	-- Skip --	xɜ	h to velar fricative; Obstruent deletion
6	ɾə	t_ʊ	

Figure 1: STAT provides an initial alignment and associated PSPs. Provided alignments and PSPs can be manually changed by a linguist, recomputed, and annotated.

4.2 Transcription Management

Whenever a transcription is to be made by linguists, a new transcription record is created, associated with a source language, and the audio file is attached to the transcription record. Once the audio file has been made available, linguists are able to use a web interface to play the audio recording and create phonetic transcriptions. The transcription management interface then allows a senior linguist to adjudicate differences between transcriptions and select an authoritative transcription.

4.3 Transcription Alignment and Analysis

Once an authoritative transcription for a speaker has been created a linguist can then compare the transcription with the previously transcribed speech of another speaker. This alignment process is the core of the system. The first stage of the comparison is to create a word and phone level alignment between the two transcriptions. The alignment is performed by our special implementation of Kondrak's phonetic alignment algorithm (Kondrak, 2000). The output from this part of the system is a complete phone-to-phone to alignment of two transcriptions. Figure 1 shows an example alignment with PSPs that a linguist is able to make adjustments to or mark correct. After alignment a linguist can perform an assessment of the speaker's speech abilities and make other notes.

To help linguists who do work with a variety of different languages and research needs, the settings for the phonemic cluster parser, phoneme distance measures, and alignment algorithm coefficient can be easily changed inside of STAT. Linguists can also control the set of constraints used for the phonological speech patterns analysis.

4.4 Phonological Speech Pattern Analysis

Once the transcription alignment has been completed, the phonological speech pattern analysis can begin. This analysis evaluates all phonetic differences between the two transcriptions under analysis. These differences are then processed by our algorithm and used to determine unique phonological speech patterns. All potential phonological speech patterns are returned to the linguist for verification. As the system encounters and stores more and more phonological speech pattern analyses for a particular language, general descriptions are made about peoples' accents from a particular language background.

5 Future Work

Our initial design of STAT uses manually determined weights of phonological features used to align transcriptions and determine phonological speech processes. In the next major release of STAT we intend to integrate automated methods to propose weight settings based on language selections.

We are currently planning on integrating a spectrographic analysis mechanism that will allow for the transcriptions to be time synchronized with the original speech sample. After this we will be investigating the integration of several speaker accent identification algorithms. We will also be investigating applications of this tool to help speech pathologists in the identification and assessment of disordered speech patterns.

6 References

Anderson-Hsieh, J., Johnson, R., & Kohler, K. (1992). The relationship between native speaker judgments of non-native pronunciation and deviance in segmentals, prosody, and syllable structure. *Language Learning,* 42, 529-555.

Bartkova, K., & Jouvet, D. (2007). On using units trained on foreign data for improved multiple accent speech recognition. *Speech Communication*, 49, 836-846.

Cucchiarini, C. (1996). Assessing transcription agreement: methodological aspects. *Clinical Linguistics & Phonetics,* 10, 131-155.

Deshpande, S., Chikkerur, S., & Govindaraju, V. (2005). Accent classification in speech. *Proceedings of the 4th IEEE Workshop on Automatic Identification Advanced Technologies*.

Kondrak, G. (2000). A new algorithm for the alignment of phonetic sequences. In *Proceedings of the First Conference on North American Chapter of the Association For Computational Linguistics* (Seattle, Washington, April 29 - May 04, 2000). ACM International Conference Proceeding Series, vol. 4. Morgan Kaufmann Publishers, San Francisco, CA, 288-295.

Shriberg, L., Hinke, R., & Trost-Steffen, C. (1987). A procedure to select and train persons for narrow phonetic transcription by consensus. *Clinical Linguistics & Phonetics,* 1, 171-189.

Morpho Challenge - Evaluation of algorithms for unsupervised learning of morphology in various tasks and languages

Mikko Kurimo, Sami Virpioja, Ville Turunen, Teemu Hirsimäki
Adaptive Informatics Research Centre
Helsinki University of Technology
FI-02015, TKK, Finland
Firstname.Lastname@tkk.fi

Abstract

After the release of the open source software implementation of Morfessor algorithm, a series of several open evaluations has been organized for unsupervised morpheme analysis and morpheme-based speech recognition and information retrieval. The unsupervised morpheme analysis is a particularly attractive approach for speech and language technology for the morphologically complex languages. When the amount of distinct word forms becomes prohibitive for the construction of a sufficient lexicon, it is important that the words can be segmented into smaller meaningful language modeling units. In this presentation we will demonstrate the results of the evaluations, the baseline systems built using the open source tools, and invite research groups to participate in the next evaluation where the task is to enhance statistical machine translation by morpheme analysis.

A proposal for a Type II Demo

1 Extended Abstract

1.1 The segmentation of words into morphemes

One of the fundamental tasks in natural language processing applications, such as large-vocabulary speech recognition (LVCSR), statistical machine translation (SMT) and information retrieval (IR), is the morphological analysis of words. It is particularly important for the morphologically complex languages, where the amount of different word forms is substantially increased by inflection, derivation and composition. The decomposition of words is required not only for understanding the sentence, but in many languages also for just representing the language by any tractable and trainable statistical model and lexicon. The manually composed rule-based morphological analyzers can solve these problems to some extent, but only a fraction of the existing languages have been covered so far, and for many the coverage of the relevant content is insufficient.

The objective of the Morpho Challenge[1] is to design and evaluate new unsupervised statistical machine learning algorithms that discover which morphemes (smallest individually meaningful units of language) words consist of. The goal is to discover basic vocabulary units suitable for different tasks, such as LVCSR, SMT and IR. In unsupervised learning the list of morphemes is not pre-specified for each language, but the optimal morpheme lexicon and morpheme analysis of all different word forms is statistically optimized from a large text corpus in a completely data-driven manner.

The evaluation of the morpheme analysis algorithms is performed both by a linguistic and an application oriented task. The analysis obtained for a long list of words is first compared to the linguistic gold standard representing a grammatically correct analysis by verifying that the morpheme-sharing word pairs are the correct ones (Kurimo et al., 2007). This is repeated in different languages and then the obtained decomposition of words is applied in state-of-the-art systems running various

[1] See http://www.cis.hut.fi/morphochallenge2009/

Proceedings of NAACL HLT 2009: Demonstrations, pages 13–16,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

NLP applications. The suitability of the morphemes is verified by comparing the performance of the systems to each other and to systems using unprocessed words or conventional word processing algorithms like stemming or rule-based decompositions.

As a baseline method in all application, we have built systems by applying the Morfessor algorithm, which is an unsupervised word decomposition algorithm developed at our research group (Creutz and Lagus, 2002) and released as open source software implementation[2].

1.2 Morphemes in Information Retrieval

In information retrieval (IR) from text documents a typical task is to look for the most relevant documents for a given query. One of the key challenges is to reduce all the inflected word forms to a common root or stem for effective indexing. From the morpheme analysis point of view this task is to decompose all the words in the query and text documents and find out those common morphemes which form the most relevant links.

In Morpho Challenge the IR systems built using the unsupervised morpheme analysis algorithms are compared in state-of-the-art CLEF tasks in Finnish, German and English (Kurimo and Turunen, 2008) using the mean average precision metric. The results are also compared to those obtained by the grammatical morphemes as well as the stemming and word normalization methods conventionally used in IR.

1.3 Morphemes in Speech Recognition

In large-vocabulary continuous speech recognition (LVCSR) one key part of the process is the statistical language modeling which determines the prior probabilities of all the possible word sequences. An especially challenging task is to cover all the possible word forms with sufficient accuracy, because any out-of-vocabulary words will not only be never correctly recognized, but also severely degrade the modeling of the other nearby words. By decomposing the words into meaningful sub-word units, such as morphemes, large-vocabulary language models can be successfully built even for the most difficult agglutinative languages, like Finnish, Estonian and Turkish (Kurimo et al., 2006b).

In Morpho Challenge the unsupervised morpheme algorithms have been compared by using the morphemes to train statistical language models and applying the models in state-of-the-art LVCSR tasks in Finnish and Turkish (Kurimo et al., 2006a). Benchmarks for the same tasks were obtained by models that utilize the grammatical morphemes as well as traditional word-based language models.

1.4 Morphemes in Machine Translation

The state-of-the-art statistical machine translation (SMT) systems are affected by the morphological variation of words at two different stages (Virpioja et al., 2007). In the first stage, the alignment of the source and target language words in a parallel training corpus and the training of the translation model can benefit from the decomposition of complex words into morphemes. This is particularly important when either the target or the source language, or both, are morphologically complex. The final stage where the target language text is generated, may also require morpheme-based models, because the large-vocabulary statistical language models are applied in the same way as in LVCSR.

In the on-going Morpho Challenge 2009 competition, the morpheme analysis algorithms are compared in SMT tasks, where the analysis is needed for the source language texts. The European Parliament parallel corpus (Koehn, 2005) is used in the evaluation. The source languages are Finnish and German and the target in both tasks is English. To obtain a state-of-the-art performance in the tasks the morpheme-based SMT will be combined with a word-based SMT using the Minimum Bayes Risk (MBR) interpolation of the N-best translation hypothesis of both systems (de Gispert et al., 2009).

1.5 Morpho Challenge 2009

As its predecessors, the Morpho Challenge 2009 competition is open to all and free of charge. The participants' are expected to use their unsupervised machine learning algorithms to analyze the word lists of different languages provided by the organizers and submit the results of their morpheme analysis. The organizers will then run the linguistic evaluations and build the IR and SMT systems and provide all the results and comparisons of the different systems. The participated algorithms and evaluation

[2] See http://www.cis.hut.fi/projects/morpho/

results will be presented at the Morpho Challenge workshop that is currently planned to take place within the HLT-NAACL 2010 conference.

Acknowledgments

The Morpho Challenge competitions and workshops are part of the EU Network of Excellence PASCAL Challenge program and organized in collaboration with CLEF. We are grateful to Mathias Creutz, Ebru Arisoy, Stefan Bordag, Nizar Habash and Majdi Sawalha for contributions in processing the training data and creating the gold standards. The Academy of Finland has supported the work in the projects *Adaptive Informatics* and *New adaptive and learning methods in speech recognition*.

References

M. Creutz and K. Lagus. 2002. Unsupervised discovery of morphemes. In *Workshop on Morphological and Phonological Learning of ACL-02*.

A. de Gispert, S. Virpioja, M. Kurimo, and W. Byrne. 2009. Minimum Bayes risk combination of translation hypotheses from alternative morphological decompositions. Submitted to *HLT-NAACL*.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit X*.

M. Kurimo and V. Turunen. 2008. Unsupervised morpheme analysis evaluation by IR experiments – Morpho Challenge 2008. In *CLEF*.

M. Kurimo, M. Creutz, M. Varjokallio, E. Arisoy, and M. Saraclar. 2006a. Unsupervised segmentation of words into morphemes - Challenge 2005, an introduction and evaluation report. In *PASCAL Challenge Workshop on Unsupervised segmentation of words into morphemes*.

M. Kurimo, A. Puurula, E. Arisoy, V. Siivola, T. Hirsimäki, J. Pylkkönen, T. Alumäe, and M. Saraclar. 2006b. Unlimited vocabulary speech recognition for agglutinative languages. In *HLT-NAACL*.

M. Kurimo, M. Creutz, and M. Varjokallio. 2007. Morpho Challenge evaluation using a linguistic Gold Standard. In *CLEF*.

S. Virpioja, J. J. Väyrynen, M. Creutz, and M. Sadeniemi. 2007. Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. In *MT Summit XI*. Denmark.

2 Script outline for the demo presentation

In this demo we will present the achievements of the Morpho Challenge 2005-2008 competition in graphs and the baseline systems for various languages developed using the Morfessor algorithm for word decomposition, IR, LVCSR and SMT. The audience will also be welcome to try their own input for these baseline systems and view the results.

The script is presented below for a poster-style and try-it-yourself on laptop demo, but it will work well as a lecture-style show, too, if needed.

In the poster we illustrate the following points:

1. Basic characteristics of the unsupervised learning algorithms and morpheme analysis results in different languages (Finnish, Turkish, German, English, Arabic) as in Table 1, demo: *http://www.cis.hut.fi/projects/morpho/*.

2. The results of the evaluations against the linguistic gold standard morphemes in different languages, see e.g. Figure 1.

3. The results of the IR evaluations and comparisons to the performance of grammatical morphemes, word-based methods and stemming in different languages, see e.g. Figure 2.

4. The results of the LVCSR evaluations with comparisons to grammatical morphemes and word-based methods, see e.g. Figure 3.

5. The call for participation in the Morpho Challenge 2009 competition where the new evaluation task is using morphemes in SMT.

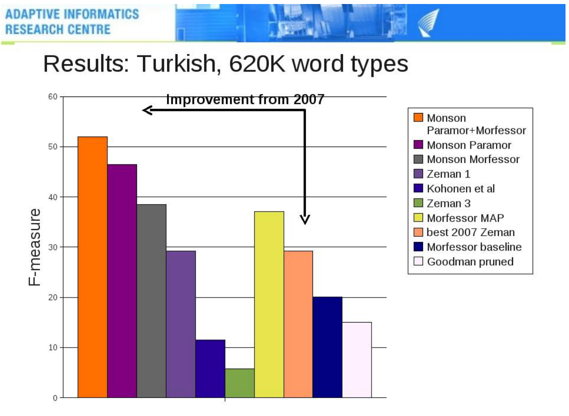


Figure 1: F-measures for the Turkish morpheme analysis.

The laptop is used to demonstrate the baseline systems we have recently developed for different tasks that are all based on unsupervised morphemes:

Example word	Morfessor analysis	Gold Standard
Finnish: linuxiin	linux +iin	linux_N +ILL
Turkish: popUlerliGini	pop +U +ler +liGini	popUler +DER_lHg +POS2S +ACC, popUler +DER_lHg +POS3 +ACC3
Arabic: AlmtHdp	Al+ mtHd +p	mut aHidap_POS:PN Al+ +SG, mut aHid_POS:AJ Al+ +SG
German: zurueckzubehalten	zurueck+ zu+ be+ halten	zurueck_B zu be halt_V +INF
English: baby-sitters	baby-+ sitter +s	baby_N sit_V er_s +PL

Table 1: Morpheme analysis examples in different languages.

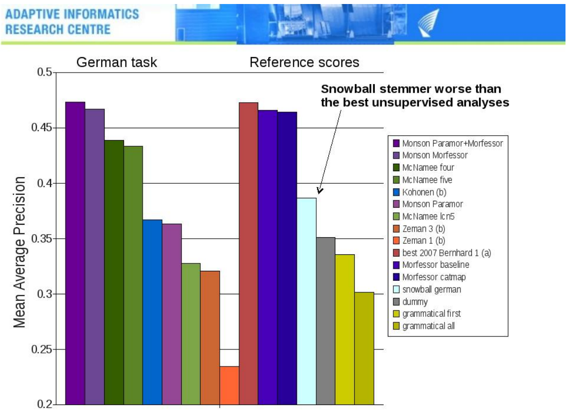


Figure 2: Precision performances for the German IR.

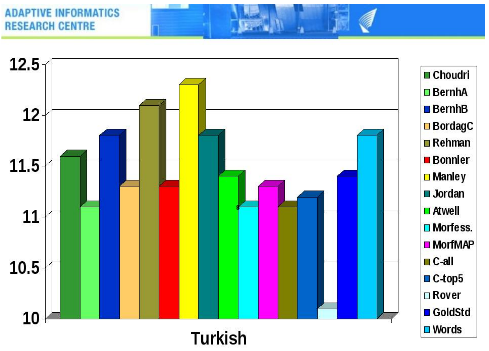
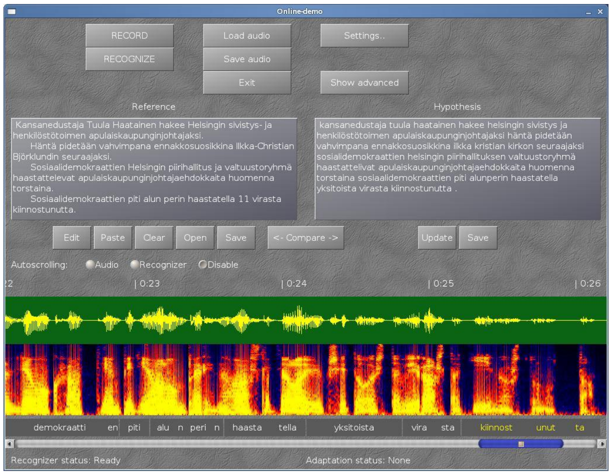


Figure 3: LVCSR error rates for the Turkish task.

1. Online LVCSR system for highly agglutinative languages, see e.g. screenshot in Figure 4.

2. Online IR system for highly agglutinative languages.

3. Online SMT system where the source language is a highly agglutinative language, see e.g. screenshot in Figure 5.



Reference

Kansanedustaja Tuula Haatainen hakee Helsingin sivistys- ja henkilöstötoimen apulaiskaupunginjohtajaksi.

Häntä pidetään vahvimpana ennakkosuosikkina Ilkka-Christian Björklundin seuraajaksi.

Sosiaalidemokraattien Helsingin piirihallitus ja valtuustoryhmä haastattelevat apulaiskaupunginjohtajaehdokkaita huomenna torstaina.

Sosiaalidemokraattien piti alun perin haastatella 11 virasta kiinnostunutta.

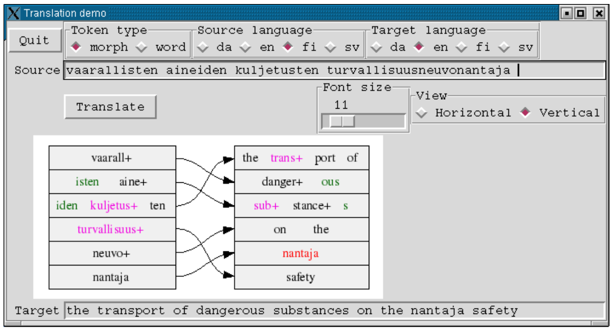
Hypothesis

kansanedustaja tuula haatainen hakee helsingin sivistys ja henkilöstötoimen apulaiskaupunginjohtajaksi häntä pidetään vahvimpana ennakkosuosikkina ilkka kristian kirkon seuraajaksi sosialidemokraattien helsingin piirihallituksen valtuustoryhmä haastattelivat apulaiskaupunginjohtajaehdokkaita huomenna torstaina sosiaalidemokraattien piti alunperin haastatella yksitoista virasta kiinnostunutta .

Recognizer status: Ready

Adaptation status: None

Figure 4: Screenshot of the morpheme-based speech recognizer in action for Finnish. An offline version can be tried in *http://www.cis.hut.fi/projects/speech/*.



Token type: ◆ morph ◇ word

Source language: ◇ da ◇ en ◆ fi ◇ sv

Target language: ◇ da ◆ en ◇ fi ◇ sv

Source vaarallisten aineiden kuljetusten turvallisuusneuvonantaja

Font size 11

View: ◇ Horizontal ◆ Vertical

Target the transport of dangerous substances on the nantaja safety

Figure 5: Screenshot of the morpheme-based machine translator in action for Finnish-English. A simplified web interface to the system is also available (please email to the authors for a link).

WordNet::SenseRelate::AllWords - A Broad Coverage Word Sense Tagger that Maximizes Semantic Relatedness

Ted Pedersen and **Varada Kolhatkar**
Department of Computer Science
University of Minnesota
Duluth, MN 55812 USA
{tpederse,kolha002}@d.umn.edu
http://senserelate.sourceforge.net

Abstract

WordNet::SenseRelate::AllWords is a freely available open source Perl package that assigns a sense to every content word (known to WordNet) in a text. It finds the sense of each word that is most related to the senses of surrounding words, based on measures found in WordNet::Similarity. This method is shown to be competitive with results from recent evaluations including SENSEVAL-2 and SENSEVAL-3.

1 Introduction

Word sense disambiguation is the task of assigning a sense to a word based on the context in which it occurs. This is one of the central problems in Natural Language Processing, and has a long history of research. A great deal of progress has been made in using supervised learning to build models of disambiguation that assign a sense to a single target word in context. This is sometimes referred to as the lexical sample or target word formulation of the task.

However, to be effective, supervised learning requires many manually disambiguated examples of a single target word in different contexts to serve as training data to learn a classifier for that word. While the resulting models are often quite accurate, manually creating training data in sufficient volume to cover even a few words is very time consuming and error prone. Worse yet, creating sufficient training data to cover all the different words in a text is essentially impossible, and has never even been attempted.

Despite these difficulties, word sense disambiguation is often a necessary step in NLP and can't simply be ignored. The question arises as to how to develop broad coverage sense disambiguation modules that can be deployed in a practical setting without investing huge sums in manual annotation efforts. Our answer is WordNet::SenseRelate::AllWords (SR-AW), a method that uses knowledge already available in the lexical database WordNet to assign senses to every content word in text, and as such offers broad coverage and requires no manual annotation of training data.

SR-AW finds the sense of each word that is most related or most similar to those of its neighbors in the sentence, according to any of the ten measures available in WordNet::Similarity (Pedersen et al., 2004).

It extends WordNet::SenseRelate::TargetWord, a lexical sample word sense disambiguation algorithm that finds the maximum semantic relatedness between a target word and its neighbors (Patwardhan et al., 2003). SR-AW was originally developed by (Michelizzi, 2005) (through version 0.06) and is now being significantly enhanced.

2 Methodology

SR-AW processes a text sentence by sentence. It proceeds through each sentence word by word from left to right, centering each content word in a balanced window of context whose size is determined by the user. Note that content words at the start or end of a sentence will have unbalanced windows associated with them, since the algorithm does not cross sentence boundaries and treats each sentence independently.

Proceedings of NAACL HLT 2009: Demonstrations, pages 17–20,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

All of the possible senses of the word in the center of the window are measured for similarity relative to the possible senses of each of the surrounding words in the window in a pairwise fashion. The sense of the center word that has the highest total when those pairwise scores are summed is considered to be the sense of that word. SR-AW then moves the center of the window to the next content word to the right. The user has the option of fixing the senses of the words that precede it to those that were discovered by SR-AW, or allowing all their senses to be considered in subsequent steps.

WordNet::Similarity[1] offers six similarity measures and four measures of relatedness. Measures of similarity are limited to making noun to noun and verb to verb comparisons, and are based on using the hierarchical information available for nouns and verbs in WordNet. These measures may be based on path lengths (path, wup, lch) or on path lengths augmented with Information Content derived from corpora (res, lin, jcn). The measures of relatedness may make comparisons between words in any part of speech, and are based on finding paths between concepts that are not limited to hierarchical relations (hso), or on using gloss overlaps either for string matching (lesk) or for creating a vector space model (vector and vector-pairs) that are used for measuring relatedness.

The availability of ten different measures that can be used with SR-AW leads to an incredible richness and variety in this approach. In general word sense disambiguation is based on the presumption that words that occur together will have similar or related meanings, so SR-AW allows for a wide range of options in deciding how to assess similarity and relatedness. SR-AW can be viewed as a graph based approach when using the path based measures, where words are assigned the senses that are located most closely together in WordNet. These path based methods can be easily augmented with Information Content in order to allow for finer grained distinctions to be made. It is also possible to lessen the impact of the physical structure of WordNet by using the content of the glosses as the primary source of information.

3 WordNet::SenseRelate::AllWords Usage

Input : The input to SR-AW can either be plain untagged text (raw), or it may be tagged with Penn Treebank part of speech tags (tagged : 47 tags; e.g., run/VBD), or with WordNet part of speech tags (wntagged: 4 tags for noun, verb, adjective, adverb; e.g., run#v). Penn Treebank tags are mapped to WordNet POS tags prior to SR-AW processing, so even though this tag set is very rich, it is used simply to distinguish between the four parts of speech WordNet knows, and identify function words (which are ignored as WordNet only includes open class words). In all cases simple morphological processing as provided by WordNet is utilized to identify the root form of a word in the input text.

Examples of each input format are shown below:

- (raw) : The astronomer married a movie star.
- (tagged) : The/DT astronomer/NN married/VBD a/DT movie_star/NN
- (wntagged) : The astronomer#n married#v a movie_star#n

If the format is raw, SR-AW will identify WordNet compounds before processing. These are multiword terms that are usually nouns with just one sense, so their successful identification can significantly improve overall accuracy. If a compound is not identified, then it often becomes impossible to disambiguate. For example, if White House is treated as two separate words, there is no combination of senses that will equal the residence of the US president, where that is the only sense of the compound White_House. To illustrate the scope of compounds, of the 155,287 unique strings in WordNet 3.0, more than 40% (64,331) of them are compounds. If the input is tagged or wntagged, it is assumed that the user has identified compounds by connecting the words that make up a compound with _ (e.g., white_house, movie_star).

In the tagged and wntagged formats, the user must identify compounds and also remove punctuation. In the raw format SR-AW will simply ignore punctuation unless it happens to be part of a compound (e.g., adam's_apple, john_f._kennedy). In all formats the upper/lower case distinction is ignored, and it is

[1] http://wn-similarity.sourceforge.net

assumed that the input is already formatted one line per sentence, one sentence per line.

SR-AW will then check to see if a stoplist has been provided by the user, or if the user would like to use the default stoplist. In general a stoplist is highly recommended, since there are quite a few words in WordNet that have unexpected senses and might be problematic unless they are excluded. For example, *who* has a noun sense of World Health Organization. *A* has seven senses, including angstrom, vitamin A, a nucleotide, a purine, an ampere, the letter, and the blood type. Many numbers have noun senses that define them as cardinal numbers, and some have adjective senses as well.

In the raw format, the stoplist check is done after compounding, because certain compounds include stop words (e.g., us_house_of_representatives). In the wntagged and tagged formats the stoplist check is still performed, but the stoplist must take into account the form of the part of speech tags. However, stoplists are expressed using regular expressions, making it quite convenient to deal with part of speech tags, and also to specify entire classes of terms to be ignored, such as numbers or single character words.

Disambiguation Options : The user has a number of options to control the direction of the SR-AW algorithm. These include the very powerful choices regarding the measure of similarity or relatedness that is to be used. There are ten such measures as has been described previously. As was also already mentioned, the user also can choose to fix the senses of words that have already been processed.

In addition to these options, the user can control the size of the window used to determine which words are involved in measuring relatedness or similarity. A window size of N includes the center word, and then extends out to the left and right of the center for $N/2$ content words, unless it encounters the sentence boundaries. If N is odd then the number of words to the left and right $(N-1)/2$, and if N is even there are $N/2$ words to the left, and $(N/2)-1$ words to the right.

When using a measure of similarity and tagged or wntagged text, it may be desirable to coerce the part of speech of surrounding words to that of the word in the center of the window of context. If this is not done, then any word with a part of speech other than that of the center word will not be included in the calculation of semantic similarity. Coercion is performed by first checking for forms of the word in a different part of speech, and then checking if there are any derivational relations from the word to the part of speech of the center word. Note that in the raw format part of speech coercion is not necessary, since the algorithm will consider all possible parts of speech for each word. If the sense of previous words has already been fixed, then part of speech coercion does not override those fixed assignments.

Finally, the user is able to control several scoring thresholds in the algorithm. The user may specify a context score which indicates a minimum threshold that a sense of the center word should achieve with all the words in the context in order to be selected. If this threshold is not met, no sense is assigned and it may be that the window should be increased.

The pair score is a finer grained threshold that indicates the minimum values that a relatedness score between a sense of the center word and a sense of one of the neighbors must achieve in order to be counted in the overall score of the center word. If this threshold is not met then the pair will contribute 0 to that score. This can be useful for filtering out noise from the scores when set to modest values.

Output : The output of SR-AW is the original text with WordNet sense tags assigned. WordNet sense tags are given in WPS form, which means word, part of speech, and sense number. In addition, glosses are displayed for each of the selected senses.

There are also numerous trace options available, which can be combined in order to provide more detailed diagnostic output. This includes displaying the window of context with the center word designated (1), the winning score for each context window (2), the non-zero scores for each sense of the center word (4), the non-zero pairwise scores (8), the zero values for any of the previous trace levels (16), and the traces from the semantic relatedness measures from WordNet::Similarity (32).

4 Experimental Results

We have evaluated SR-AW using three corpora that have been manually annotated with senses from WordNet. These include the SemCor corpus, and

Table 1: SR-AW Results (%)

	2			5			15		
SC	P	R	F	P	R	F	P	R	F
lch	56	13	21	54	29	36	52	35	42
jcn	65	15	24	64	31	42	62	41	49
lesk	58	49	53	62	60	61	62	61	61
S2	P	R	F	P	R	F	P	R	F
lch	48	10	16	50	24	32	48	31	38
jcn	55	9	15	55	21	31	55	31	39
lesk	54	44	48	58	56	57	59	59	59
S3	P	R	F	P	R	F	P	R	F
lch	48	13	20	49	29	37	48	35	41
jcn	55	14	22	55	31	40	53	38	46
lesk	51	43	47	54	52	53	54	53	54

the Senseval-2 and Senseval-3 corpora. SemCor is made up of more than 200,000 words of running text from news articles found in the Brown Corpus. The Senseval data sets are each approximately 4,000 words of running text from Wall Street Journal news articles from the Penn Treebank. Note that only the words known to WordNet in these corpora have been sense tagged. As a result, there are 185,273 sense tagged words in SemCor, 2,260 in Senseval-2, and 1,937 in Senseval-3. We have used versions of these corpora where the WordNet senses have been mapped to WordNet 3.0[2].

In Table 4 we report results using Precision (P), Recall (R), and F-Measure (F). We use three window sizes in these experiments (2, 5, and 15), three WordNet::Similarity measures (lch, jcn, and lesk),and three different corpora : SemCor (*SC*), Senseval-2 (*S2*), Senseval-3 (*S3*). These experiments were carried out with version 0.17 of SR-AW.

For all corpora we observe the same patterns. The lesk measure tends to result in much higher recall with smaller window sizes, since it is able to measure similarity between words with any parts of speech, whereas lch and jcn are limited to making noun-noun and verb-verb measurements. But, as the window size increases so does recall. Precision continues to increase for lesk as the window size increases. Our best results come from using the lesk measure with a window size of 15. For SemCor this results in an F-measure of 61%. For Senseval-2 it results in an F-measure of 59%, and for Senseval-3 it results in an F-measure of 54%. These results would have ranked 4th of 22 teams and 15th of 26 in the respective Senseval events.

A well known baseline for all words disambiguation is to assign the first WordNet sense to each ambiguous word. This results in an F-measure of 76% for SemCor, 69% for Senseval-2, and 68% for Senseval-3. A lower bound can be established by randomly assigning senses to words. This results in an F-Measure of 41% for SemCor, 41% for Senseval-2, and 37% for Senseval-3. This is relatively high due to the large number of words that have just one possible sense (so randomly selecting will result in a correct assignment). For example, in SemCor approximately 20% of the ambiguous words have just one sense. From these results we can see that SR-AW lags behind the sense one baseline (which is common among all words systems), but significantly outperforms the random baseline.

5 Conclusions

WordNet::SenseRelate::AllWords is a highly flexible method of word sense disambiguation that offers broad coverage and does not require training of any kind. It uses WordNet and measures of semantic similarity and relatedness to identify the senses of words that are most related to each other in a sentence. It is implemented in Perl and is freely available from the URL on the title page both as source code and via a Web interface.

References

J. Michelizzi. 2005. Semantic relatedness applied to all words sense disambiguation. Master's thesis, University of Minnesota, Duluth, July.

S. Patwardhan, S. Banerjee, and T. Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 241–257, Mexico City, February.

T. Pedersen, S. Patwardhan, and J. Michelizzi. 2004. Wordnet::Similarity - Measuring the relatedness of concepts. In *Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 38–41, Boston, MA.

[2] http://www.cse.unt.edu/~rada/downloads.html

Author Index