Speech Translation

Stephan Vogel
19 January 2011
Overview

- Some speech translation projects
- Some examples of speech translation systems
- Speech translation pipeline – some considerations
Speech Translation
Projects and Demos
A (Personal) Speech Translation Journey
A (Personal) Speech Translation Journey

- Verbmbil
- EuTrans
- Nespole
- Babylon
- Str-Dust
- TC-Star
- GALE
- TransTac

Limited Domain
Unlimited Domain

Machine Translation
A (Personal) Speech Translation Journey

1996
Verbmobil

1997
EuTrans

1998
1999

2000

Nespolo

2001
Babylon

2002

2003

2004

2005

Str-Dust

2006

TC-Star

2007

GALE

2008

TransTac

TIDES

Statistical Phrase-based

Interlingua

Machine Translation
The Verbmobil Project

The Quick and Dirty 😊 = best in the final evaluation
Verbmobil Highlights

- Multiple recognizers (languages and systems)
- Multiple translation engines
  - Transfer, with deep syntactic/semantic analysis
  - Dialog-Act-based
  - Example based
  - Statistical
- System combination for translation engines
  - Did not work
- Prosodic annotation (probabilistic, on lattices)
  - Segment boundaries on various levels
  - Question
  - Accent
  - Actually used by translation modules
- Disfluency detection and removal (on lattices)
Nespole!

- Project
  - C-Star partners
  - Interlingua-based translation (English, German, Italian)
  - Travel domain

- Outside of project: is SMT possible on very small corpora?

Train: 3182 parallel speech dialog units (sentences)

<table>
<thead>
<tr>
<th>Language</th>
<th>English</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokens</td>
<td>15572</td>
<td>14992</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>1032</td>
<td>1338</td>
</tr>
<tr>
<td>Singletons</td>
<td>404</td>
<td>620</td>
</tr>
</tbody>
</table>

Test: 70 Parallel SDUs

<table>
<thead>
<tr>
<th></th>
<th>German</th>
<th>Reference A</th>
<th>Reference B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokens</td>
<td>437</td>
<td>610</td>
<td>607</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>183 (45 oov)</td>
<td>165</td>
<td>160</td>
</tr>
</tbody>
</table>
Evaluation

• Human Scoring
  • Good, Okay, Bad (c.f. Nespolo evaluation)
  • Collapsed into a „human score“ on [0,1] (good = 1.0, okay = 0.5)

• Bleu Score
  • Average of N-gram precisions from (1..N), typically N=3 or 4
  • Penalty for short translations to substitute for recall measure
  • Numbers ranging 0.0 ... 1.0, higher is better

<table>
<thead>
<tr>
<th></th>
<th>Good</th>
<th>Okay</th>
<th>Bad</th>
<th>Score</th>
<th>Bleu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>SMT</td>
<td>127</td>
<td>80</td>
<td>205</td>
<td>0.40</td>
</tr>
<tr>
<td></td>
<td>IF</td>
<td>77</td>
<td>104</td>
<td>227</td>
<td>0.32</td>
</tr>
<tr>
<td>Speech</td>
<td>SMT</td>
<td>95</td>
<td>83</td>
<td>227</td>
<td>0.34</td>
</tr>
<tr>
<td></td>
<td>IF</td>
<td>64</td>
<td>101</td>
<td>243</td>
<td>0.28</td>
</tr>
</tbody>
</table>

• Results
  • SMT works (as good) with very small training data
  • Both systems pretty useless (OOV!!!)
Translation System for Travel Domain

Central question:

How to build a translation system which can be used when you are “On the Run”
Option I

- Travel with powerful computers
Option II

- Communicate with a powerful server via wireless*

* Problem: Wireless networks are not always available

---

Option III: PanDoRA

- Smart engineering of the SMT system so that it can run on hand-held devices
- All components on device
  - No dependency on connectivity
PanDoRA: System Architecture
Challenges

- **Memory**
  - SMT systems require several GB RAM on PCs
  - Hand-held devices have very small dynamic memory
  - 64MB available on iPaq 2750; shared with ASR and TTS

- **CPU**
  - Speech translator requires real-time translations
  - CPUs on hand-held devices are usually slow (e.g. 600M Hz) and
  - There are no coprocessors for floating-point arithmetic
Solutions

- Compact data structures
- Integerized computation
- Efficient decoding
- Minimum on-device computation
Compact Data Structure

- Each phrase is represented by its <location, length> in the corpus rather than a string of words.
- A translation pair is represented by <src phrase index, tgt phrase index, score>.

Arabic corpus:
- احتفال المدرسة
- احتفال المدرسة عقد

English corpus:
- this year's school festival is hold on September.
- ....
- .... there will be a doll's festival....
- ...festive....
Compact Data Structure

- Data records now have fixed size in bytes, independent of the length of a phrase.
- Sorted list of all model data is stored on external memory card
  - Much larger capacity than SDRAM. e.g. 2GB SD card
  - Saves SDRAM for decoding
- Search for matched phrases by binary search in the sorted list
- Access the information by 'seek'ing the record from a file by record id.
  - e.g. getPhrase(id); getTranslationOfPhrase(id)...
- With compact data structure a 5.6 million entry phrase table requires 65MB on disk
- Specification for current implementation
  - 64K Vocabulary for each language
  - Up to 256 million unique src/tgt phrases allowed
  - Up to 4 billion phrase pairs allowed
**Integerized Computing**

- SMT systems are “translating by numbers”
- Hand-held devices usually do not have numerical co-processors -> slow with floating points calculations
- PanDoRA's solution: integerized computing
  - Convert all probabilities to cost = -log(prob)
  - Quantize the cost to integer bins between [0, 4095]
  - Probabilities put into the same bin are considered equal during decoding.

<table>
<thead>
<tr>
<th>TM</th>
<th>LM</th>
<th>BLEU</th>
<th>NIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Float</td>
<td>Float</td>
<td>19.87</td>
<td>8.03</td>
</tr>
<tr>
<td>Int</td>
<td>Float</td>
<td>19.82</td>
<td>7.99</td>
</tr>
<tr>
<td>Float</td>
<td>Int</td>
<td>19.94</td>
<td>8.03</td>
</tr>
<tr>
<td>Int</td>
<td>Int</td>
<td>19.93</td>
<td>8.04</td>
</tr>
</tbody>
</table>
Minimum On-device Computation

- Log-linear model in SMT
  - Multiple model features
    \[ P(e^I_1 \mid f^J_1) = \frac{\exp(\sum_{m=1}^{M} \lambda_m \phi_m (e^I_1, f^J_1))}{Z} \]

- Off-line model optimization
  - Minimum Error (MER) training off-line
  - Combine weighted translation model scores into a single score
Decoding

- Inverted Transduction Grammar (ITG) style decoding [bottom-up]
  - Translation as parsing
  - Combine adjacent partial hypotheses either straight or inverted to create new hypothesis covering the combined source range.
  - Allows for long distance reordering. Effective for language pairs with dramatically different word ordering. E.g. Japanese/English
  - Language model adjusted on the boundary words.
**Speechalator**

- **Graphical User Interface**
  - Two windows, one per language, fixed layout
  - Simple push-to-talk recording (soft and hard buttons)
  - Color-coded progress bars (separate for input, ASR, MT)

- **System Features**
  - Back translation for result verification
  - Edit text using virtual keyboard
  - Re-synthesize input, re-speak output
  - Optional: no-click translation or verify before translation
  - Logging function
Travel Domain Translation Demo
From Limited towards Unlimited

Limited Domain

Unlimited Domain

Verbmobil
EuTrans
Nespole
Babylon
TC-Star
Str-Dust
GALE
TransTac
TC-Star

- Technology and Corpora for Speech to Speech Translation
- Financed by European Commission within the Sixth Program
- Research in all core technologies for Speech-to-Speech Translation

Long-term research goals of the project:
- Effective Spoken Language Translation of unrestricted conversational speech on large domains of discourse.
- Speech recognition able to adapt to and perform reliably under varying speaking styles, recording conditions, and for different user communities.
- Effective integration of speech recognition and translation into a unique statistically sound framework. (Lattice translations as explicit goal)
- General expressive speech synthesis imitating the human voice. (Development of new models for prosody, emotions and expressive speech.)
TC-Star Evaluations

- **EPPS:** European Parliamentary Plenary Sessions
  - EU >20 official languages
  - Speeches given in native language
  - Simultaneous interpretation (verbatim transcript)
  - Final text edition for publication (often significantly edited)

- **TC-Star Evaluations**
  - Show-case for potential translation (interpretation) service in the European Parliament
  - Spanish-to-English and English-to-Spanish
    - Verbatim transcript
    - ASR output (1-best, lattice)
    - Final text edition
  - Also Chinese-to-English
Lecture Translator

- Speech recognition
  - Large vocabulary
  - Continuous speech
  - Fair amount of disfluencies

- Translation
  - English -> Spanish
  - English -> German
  - Trained on EPPS (European parliament) data
  - English -> Arabic (in the works)
  - Additional LM data for adaptation to domain

- Applications
  - Translating European Parliament speeches
  - Translating lectures, e.g. on speech and translation technology
Lecture Translation Demo
MORNING THANK YOU VERY MUCH IT IS VERY NICE TO HAVE YOU HERE IT IS A PLEASURE TO TALK TO YOU ABOUT CERTAIN TECHNOLOGICAL ADVANCES THAT WE WOULD LIKE TO SHOW YOU ON THIS DAY FOR THE FIRST TIME WE'D LIKE TO INTRODUCE TO YOU SPEECH

mañana muchas gracias es muy bonito decir que usted aquí es un placer le voy a hablar algunas tecnológicos nos gustaría mostrarle en este día de la primera vez nos gustaría introducir les

heute vormittag vielen dank es ist sehr e zu haben wenn sie in nizza hier es ist eine freude sprechen sie über bestimmte technische daß gerne zeigen würden wir fortschritte an diesem tag zum ersten mal wir möchten ihnen gerne vorstellen

الصباح اشكركم كثيرا جدا ان ه والتحاور هل هو هنا ويسرني ان التحدث عن لكم بعض التكنولوجي نود أن نبين لكم على
Next Grand SLT Challenge: Multilingual Meetings

- Several meeting projects: AMI and AMIDA, CHIL, ICSI, ...
- So far not translation involved, but would be a natural extension
  - Participants have often different native languages
  - There could be a real need, even more than for lecture translation

Interesting research issues

- Latency in translation
  - How to achieve short latency, how does is affect translation quality
  - What is ‘best’ latency for user
- How to integrate external knowledge sources
  - Slides, notes from previous meetings
  - Online search for model adaptation
- Learning from human interpreters
  - Strategies to translate despite of missing information
- Presenting the translations
  - Audio or video display
  - How to deal with cross talk and parallel discussions
Speech Translation –
The Pipeline
Speech-to-Speech Translation

- Speech (acoustic signal) in → speech recognition (ASR)
- Speech (acoustic signal) out → speech synthesis (TTS)
- In between: text to text (kind of) → Translation (MT)
Connecting the Modules: ASR - MT

What is propagated from recognizer to translator?

- First best hypothesis, n-best list, entire lattice
- Additional annotations: internal scores of recognizer, acoustic, language model, confidence scores
- Prosodic information
Connecting the Modules: MT - TTS

What is propagated from translation to synthesis?
- Sequence of words
- How important is case information, punctuation?
- Acoustic information: gender of speaker, f0, accents
- Some of this information attached to specific words -> alignment
Some Research Topics

- Tight coupling between recognition and translation
  - Lattice translation
  - Using ASR features (LM, acoustic scores, confidences) in MT decoder
  - End-to-End optimization

- Segmentation and restoring punctuation

- Short latency translation

- Disfluency detection and removal
Lattice Translation: Motivation

- Lattice translation is used for tighter coupling between speech recognition and translation
  - Use alternative word sequences
  - Lattice is more efficient than n-best list
  - Word lattice has lower word error rate than 1-best hypothesis
    -> perhaps we find a better path in translation
  - Perhaps a different path gives better translation even when more recognition errors
  - Decouple processing steps, yet use relevant information for end-to-end optimization

- Lattices can be used to encode alternatives arising from other knowledge sources
  - Disfluency annotation: add edges to lattice to skip over disfluencies
  - Add synonyms and paraphrases
  - Add morphological variants, esp. for unknown words
  - Allow different word segmentation (e.g. For Chinese)
  - Partial word reordering
Requirement: Using ASR Features

- Problem when translating lattices
  - Perhaps bad path is chosen because it is “easier” to translate
  - Shorter paths are more likely to be selected

- Solution: Use Recognizer Features
  - Acoustic scores: local to source word
  - Language model scores: non-local
    - Expansion of the word lattice for different histories
    - Calculation on the fly, i.e. in the SMT decoder
    - Affected by word reordering
  - Confidence scores

- Other features might be interesting
  - Segmentation: probabilities for boundary after each word
  - Consolidation modules: probabilities for new edges

- Goal: overall optimization of end-to-end system
Requirement: Word Reordering

- Need to know which source words have already been translated
  - Don’t want to miss some words
  - Don’t want to translate words twice
  - Can compare hypotheses which cover the same words

- Coverage vector to store this information
  - For ‘small jumps ahead’: position of first gap plus short bit vector
  - For ‘small number of gaps’: array of positions of uncovered words
  - For ‘merging neighboring regions’: left and right position

- All simple and easy for sentences as input – but does this work for lattices?
Coverage Information – Confusion Nets

- Empty edges can be eliminated by expanding all incoming edges to next node
- Structure of confusion net is same as structure of translation lattice after inserting word and phrase translations
- Works fine for all reordering strategies
Coverage Information – General Lattices

- Reordering strategy 1 (sliding window) is somewhat messy
  - Can be simulated with strategy 2, so don’t bother
- Strategies 2 (IBM) and 3 (ITG) pose no additional problem
  - For 2: Recombine hyps when ending in same node and having same untranslated edges
  - For 3: Recombine hyps which have same left and right node
Using Source Language Model

- Edges for target words/phrases need to be linked to source word edges

- Easiest in reordering strategy 3
  - Run over all words attached to source edges
  - Run over all words in target edges
  - One side could have the two joint segments swapped

- In reordering strategy 2
  - Calculate source LM prob when leaving word/phrase untranslated
  - No future cost estimate needed for source LM
  - Calculate target LM prob when filling the gap
Lattice Translation Results

- A number of papers
  - General word graphs
  - Confusion nets
  - N-best lists (with N=10...100) converted into lattices
- Improvements (small and smaller) reported
- General Picture?
  ... there is none
  - Worked last time year - not this year (Wade Shen IWSLT 05 and 06)
  - Works only on small lattices - works on large lattices (Saleem 04, Matusov 07)
  - Works only for mid-range WER – works best for low WER (but your WER ranges might be different from mine)
  - Nice improvements - only small improvements
  - Translation system selects path with lower WER – selects paths with higher WER (Lane, Paulik, inhouse results)

- Problem: our models are too weak to select the few better paths in a word lattice among the zillions of worse paths
Segmentation

- Wrong segmentation introduces errors
  - Reordering over sentence boundaries
  - Loosing LM context
  - Loosing phrase matches

- Segmentation approaches
  - Simply use pause longer then fixed threshold
  - Add LM
  - Add prosodic features

- General picture
  - Wrong segmentation hurts
  - Longer or shorter segments better?
    - Conflicting results
    - Longer seems better
  - Looking at phrase table seems to help
Punctuation Recovery

Different approaches:

- Use prosodic features as punctuation substitute
- Hidden event LM
  - On Target: Delete all punctuation from MT training corpus, restore punctuation on target side through HELM
  - On Source: Restore punctuation on source side and used MT trained on corpus with punctuation
- Asymmetric translation model
  - Remove punctuation on source side of training data
  - Or from source side of phrase table
  - Generate punctuation on target side
Punctuation Recovery: Some Results

Source: Punctuation recovery applied to input source
Target: Punctuation recovery applied after translation (secondary-punctuation ignored during translation)
SMT: Punctuation recovered during translation
Manual: Input source manually annotated with punctuation
Low Latency Decoding

- Speech recognizer advances in small increments
  - Early decision can lead to errors
  - Despite LM horizon of only a few words

- Translation relies on
  - LM: need sufficient context
  - Phrase pairs: i.e. we need to trail a couple of words behind recognizer to allow for longer words
  - Word reordering: may require even longer trailing (but in practice reordering window restricted to <= 6 words)

- In decoding
  - Output partial translation
  - Use segmentation (acoustic, linguistic)

- Now also used in Google’s text translation
Translating Spoken Language

- Spoken language differs from written language
  - Vocabulary
  - Style
  - Sentence structure
  - ... 

- How to deal with this
  - Specific translation models
  - Training on specific data – hardly available
  - Adaptation approaches
Speech – It’s nasty !-

- Spoken language is noisy
  - Repetitions
  - Corrections
  - Filler words (interjections)
  - Filled pauses (hesitations)
  - Word breaks
  - False starts

- For ASR: need training data, adaptation

- For translation: difficult to get sufficient training data
  - Typically much larger text corpora than transcribed data
Clean up!

- Want to make it look more like text input
- Remove fillers, repetitions, repaired part, etc
  - Deliver one sentence which is more like written language
  - Approach cleaning by summarization: extract content
- Hard decision: errors can not be undone in MT module
Augment to Improve - Or Mess up!?

- To avoid hard decisions: Add alternatives
  - Create an N-best list or lattice -> wider pipelines
  - Works on 1-best hypothesis but also on word lattice
  - Higher computation load for subsequent MT module

- Will the better alternative be selected in translation?
  - How to score the new paths?
  - How to use these scores in the translation system?
  - We want: more alternatives to find a better one :-)
  - Problem: more ways to make errors, more errors are made :-(

User Interface Problems

- Output of speech translation systems is difficult to understand
  - Translation quality
  - Missing or incorrect segmentation: Really big problem for user

- Text display
  - Difficult to ‘parse’ the system output
  - Text display without punctuation is difficult to read
  - Wrong line-breaks have to be (mentally) undone

- Speech synthesis
  - Rather monotone and unstructured
  - Overlay with original speech
  - Real-time processing by user is hardly possible
Summary

- Speech translation: from domain limited to domain unlimited
  - Domain limited possible on hand-held devices
  - Domain unlimited translation in real-time
- Different translation approaches for speech translation
  - Interlingua-based, esp. for dialog systems
  - Phrase-based statistics also for unlimited domain translation
- Special speech translation problems
  - Segmentation of ASR output
  - Disfluencies and ill-formed utterances
  - Wider pipelines: e.g. lattice translation, propagation of features