Principles of Machine Translation
Research and Development

Alon Lavie
Language Technologies Institute
Carnegie Mellon University

11-731: Machine Translation
January 24, 2007

January 24, 2007
11-731: MT

Machine Translation:
Where are we today?

- Age of Internet and Globalization – great demand for MT:
  - Multiple official languages of UN, EU, Canada, etc.
  - Documentation dissemination for large manufacturers (Microsoft, IBM, Caterpillar)
- Economic incentive is still primarily within a small number of language pairs
- Some fairly good commercial products in the market for these language pairs
  - Primarily a product of rule-based systems after many years of development
- Pervasive MT between most language pairs still non-existent and not on the immediate horizon

Core Challenges of MT

- **Ambiguity:**
  - Human languages are highly ambiguous, and differently in different languages
  - Ambiguity at all “levels”: lexical, syntactic, semantic, language-specific constructions and idioms
- **Amount of required knowledge:**
  - At least several 100k words, at least as many phrases, plus syntactic knowledge (i.e. translation rules).
  - How do you acquire and construct a knowledge base that big that is (even mostly) correct and consistent?

How to Tackle the Core Challenges

- **Manual Labor:** 1000s of person-years of human experts developing large word and phrase translation lexicons and translation rules.
  - Example: Systran’s RBMT systems.
- **Lots of Parallel Data:** data-driven approaches for finding word and phrase correspondences automatically from large amounts of sentence-aligned parallel texts.
  - Example: Statistical MT systems.
- **Learning Approaches:** learn translation rules automatically from small amounts of human translated and word-aligned data.
  - Example: AVENUE’s XFER approach.
- **Simplify the Problem:** build systems that are limited-domain or constrained in other ways.
  - Examples: CATALYST, NESPOLE!

State-of-the-Art in MT

- What users want:
  - General purpose (any text)
  - High quality (human level)
  - Fully automatic (no user intervention)
- We can meet any 2 of these 3 goals today, but not all three at once:
  - FA HQ: Knowledge-Based MT (KBMT)
  - FA GP: Corpus-Based (Example-Based) MT
  - GP HQ: Human-in-the-loop (efficiency tool)
Types of MT Applications:

- **Assimilation**: multiple source languages, uncontrolled style/topic. General purpose MT, no semantic analysis. (GP FA or GP HQ)
- **Dissemination**: one source language, controlled style, single topic/domain. Special purpose MT, full semantic analysis. (FA HQ)
- **Communication**: Lower quality may be okay, but system robustness, real-time required.

Analysis and Generation

**Main Steps**

**Analysis**:
- Morphological analysis (word-level) and POS tagging
- Syntactic analysis and disambiguation (produce syntactic parse-tree)
- Semantic analysis and disambiguation (produce symbolic frames or logical form representation)
- Map to language-independent Interlingua

**Generation**:
- Generate semantic representation in TL
- Sentence Planning: generate syntactic structure and lexical selections for concepts
- Surface-form realization: generate correct forms of words

Statistical MT (SMT)

- Proposed by IBM in early 1990s: a direct, purely statistical, model for MT
- Statistical translation models are trained on a sentence-aligned parallel bilingual corpus
  - Train word-level alignment models
  - Extract phrase-to-phrase correspondences
  - Apply them at runtime on source input and "decode"
- Attractive: completely automatic, no manual rules, much reduced manual labor
- Main drawbacks:
  - Effective only with large volumes (several mega-words) of parallel text
  - Broad domain, but domain-sensitive
  - Still viable only for small number of language pairs!
- Impressive progress in last 5 years
  - Large DARPA funding programs (TIDES, GALE)
  - Lots of research in this direction
  - GIZA++, Pharaoh, CAIRO

Direct Approaches

- No intermediate stage in the translation
- First MT systems developed in the 1950’s-60’s (assembly code programs)
  - Morphology, bi-lingual dictionary lookup, local reordering rules
  - “Word-for-word, with some local word-order adjustments”
- Modern Approaches: EBMT and SMT

Approaches to MT: Vauquois MT Triangle

Interlingua

Analysis

Transfer

Generation

Automatic: Alon Lavie

Direct

No Error in Alon Lavie

EBMT Paradigm

New Sentence (Source)

Yesterday, 200 delegates met with President Clinton.

Matches to Source Found

Yesterday, 200 delegates met with President Clinton.

Premulti-100

Premulti-100

EBMT Paradigm

Translational Sentence (Target)

Gestern trafen sich 200 Abgeordnete hinter verschlossenen Türen mit... Schwierigkeiten mit...
Transfer Approaches

• **Syntactic Transfer:**
  - Analyze SL input sentence to its syntactic structure (parse tree)
  - Transfer SL parse-tree to TL parse-tree (various formalisms for specifying mappings)
  - Generate TL sentence from the TL parse-tree

• **Semantic Transfer:**
  - Analyze SL input to a language-specific **semantic representation** (i.e., Case Frames, Logical Form)
  - Transfer SL semantic representation to TL semantic representation
  - Generate syntactic structure and then surface sentence in the TL

Main Advantages and Disadvantages:

• **Syntactic Transfer:**
  - No need for semantic analysis and generation
  - Syntactic structures are general, not domain specific → Less domain dependent, can handle open domains
  - Requires word translation lexicon

• **Semantic Transfer:**
  - Requires deeper analysis and generation, symbolic representation of concepts and predicates → difficult to construct for open or unlimited domains
  - Can better handle non-compositional meaning structures → can be more accurate
  - No word translation lexicon – generate in TL from symbolic concepts

Knowledge-based Interlingual MT

• The classic "deep" Artificial Intelligence approach:
  - Analyze the source language into a detailed symbolic representation of its meaning
  - Generate this meaning in the target language

• "Interlingua": one single meaning representation for all languages
  - Nice in theory, but extremely difficult in practice:
    • What kind of representation?
    • What is the appropriate level of detail to represent?
    • How to ensure that the interlingua is in fact universal?

Interlingua versus Transfer

• With interlingua, need only N parsers/generators instead of N² transfer systems:

Multi-Engine MT

• Apply several MT engines to each input in parallel
• Create a combined translation from the individual translations
• Goal is to combine strengths, and avoid weaknesses.
• Along all dimensions: domain limits, quality, development time/cost, run-time speed, etc.
• Various approaches to the problem

Speech-to-Speech MT

• Speech just makes MT (much) more difficult:
  - Spoken language is messier
  - False starts, filled pauses, repetitions, out-of-vocabulary words
  - Lack of punctuation and explicit sentence boundaries
  - Current Speech technology is far from perfect
  - Need for speech recognition and synthesis in foreign languages
  - Robustness: MT quality degradation should be proportional to SR quality
  - Tight Integration: rather than separate sequential tasks, can SR + MT be integrated in ways that improves end-to-end performance?
Major Sources of Translation Problems

- **Lexical Differences:**
  - Multiple possible translations for SL word, or difficulties expressing SL word meaning in a single TL word

- **Structural Differences:**
  - Syntax of SL is different than syntax of the TL: word order, sentence and constituent structure

- **Differences in Mappings of Syntax to Semantics:**
  - Meaning in TL is conveyed using a different syntactic structure than in the SL

- **Idioms and Constructions**

Main Research Challenges

- **MT systems are complex:**
  - Design and engineering of complex set of components

- **Resources:**
  - What data and linguistic resources are required, are available, and how do we acquire them?
  - Human resources: language experts, MT experts
  - Computational resources

- **Task Requirements and Constraints:**
  - Where is the system going to run?
  - Who are the clients/users?
  - Real-time or offline?

Design and Engineering Issues

- Breakdown into a sequence of components
- Pipeline architecture vs. more integrated interaction between components
- Representation formats

System Architecture Design and Engineering

- Even the direct approaches are very complex systems that require challenging engineering:
  - Analysis components: morphological analyzer, word segmentors, tokenizers
  - Translation components: word-to-word and phrase-to-phrase transducers
  - Decoders and target language generator components
- Deeper-level approaches are even more complex:
  - Syntactic and semantic parsers and generators

Example: CMU XFER MT System
Pipeline VS Tight Integration
• Sequence of components is necessary
  – Modularizes the system, breaks it down into meaningful components for development
• Classic Pipeline: each component takes one input and produces a single selected output
  – Advantages: simplifies intermediate representations and component integration
  – Main disadvantage: cumulative error – any mistakes made by one component cannot easily be corrected down the line → errors accumulate
• Tight Integration: delay resolution of ambiguities until the best resources are available for resolving them
  – Components receive multiple possible inputs and produce multiple possible outputs
  – Requires complex data structures for “passing along” the various possible outputs

Representation Formats
• Text Strings:
  – Most common representation for SL input and TL output
  – Are often “annotated” with additional information: segmented word boundaries, identified “tokens”, Named Entities, etc.
• Structure Representations:
  – Parse trees, dependency structures, etc.
• Lattices: very commonly used form for compactly representing a collection of overlapping alternative partial or complete input and/or output structures

Lattice Representations
• Input word: B$WRH

|-----B-----|$WR|--H--|
|--B--|-H--|--$WRH---|

MEMT Chart Lattice

Resource Acquisition
• Types of resources used in MT:
  – Monolingual linguistic resources
  – Bilingual/Multilingual resources
• Finding them
• Building them

Monolingual Resources
• Raw-form resources:
  – Large monolingual corpora
• Processed resources:
  – Monolingual lexicons
  – Language Models
Language Modeling for MT

- A technique originally "stolen" from Speech Recognition, a direct consequence of the "noisy channel" model
  - $P(E|F) \sim P(F|E) \times P(E)$
  - Find the sequence of words $E$ that maximizes the above, using search
- Attempts to model the statistics of English word sequences
- Most common used model: trigram models
- Trigram example: $P(\text{Bush} | \text{George W.})$
- Statistical LM focus is on accurately estimating these probabilities from data and dealing with data sparsity
- Does not directly tackle the challenge of discriminating between the alternatives proposed by the MT decoder: trigrams do not discriminate well between good and bad translations.

Parallel Corpora

- Most attractive and valuable "raw-form" resource for many of today’s MT approaches:
  - Consists of “parallel” versions of the same text in multiple languages (at least two)
  - Most commonly aligned at the sentence-level
  - Used for extracting bilingual word and phrase lexicons (SMT, EBMT, XFER MT) phrase-to-phrase mappings (SMT), example base (EBMT), or learning structural correspondences (Syntax-driven MT approaches)

Parallel Corpora

- Where do we find/get them?
  - Produced “naturally” by entities such as the UN, EU, Canadian Parliament, etc.
  - “Comparable” corpora
  - Construct a “targeted” parallel corpus
    - The CMU Elicitation corpus
- Challenges:
  - Sentence Alignment
  - Quality: Is the corpus truly parallel?
  - Coverage

Computational Resources

- Modern MT often requires vast computational resources for both training and runtime:
  - Fast machines and large amounts of memory
  - Training: word alignment, phrase-to-phrase and transfer rule mappings
  - Extracted models can be enormous! Storing them and retrieving them...
  - Decoding and LMs can require very large amounts of memory

Task Requirements and Constraints

- Assimilation or dissemination scenario?
- Limited or broad domain?
- Where is the system going to run?
  - Server, laptop, PDA?
- Who are the clients/users?
- Real-time or offline?

MT Development Principles

1. Analyze and define the task
2. Design the system architecture
3. Acquire the necessary resources
4. Training/development of system components
5. Full prototype integration
6. Development/Testing Cycle
Development/Testing Cycle

1. Create a "development set" and a "testing set" of data
2. Run MT system on development set and assess performance
3. Error Analysis
4. Updating of system components
5. Retest performance on development data
6. Test performance on test data
7. Go back to step 3... (when do we stop?)

Error Analysis

- **Goal**: identify the causes of the most meaningful and important sources of errors in the system and correct/improve them
  1. Collect and aggregate errors
     - what are the major types of translation errors?
     - Collect statistics on these errors: how frequent are they? How important are they?
  2. Blame Assignment: what components or processes are responsible for each of the identified errors?
  3. Correction Strategies: what is required in order to correct/improve the underlying source of the error?
  4. Development work...

Questions...