Machine Translation

J&M’s Chapter 25
Machine Translation

• Automatically translate one natural language into another.

Mary didn’t slap the green witch.

Maria no dió una bofetada a la bruja verde.
Ambiguity Resolution is Required for Translation

• Syntactic and semantic ambiguities must be properly resolved for correct translation:
  – “John plays the guitar.” → “John gitar minavazad.”
  – “John plays soccer.” → “John footbal bazi mikonad.”

• An early MT system gave the following results when translating from English to Russian and then back to English:
  – “The spirit is willing but the flesh is weak.” ⇒ “The liquor is good but the meat is spoiled.”
  – “Out of sight, out of mind.” ⇒ “Invisible idiot.”
Word Alignment

• Shows mapping between words in one language and the other.

Mary didn’t slap the green witch.

Maria no dió una bofetada a la bruja verde.
Translation Quality

- Achieving literary quality translation is very difficult.
- Existing MT systems can generate rough translations that frequently at least convey the gist of a document.
- High quality translations possible when specialized to narrow domains, e.g., weather forecasts.
- Some MT systems used in *computer-aided translation* in which a bilingual human *post-edits* the output to produce more readable accurate translations.
- Frequently used to aid *localization* of software interfaces and documentation to adapt them to other languages.
Linguistic Issues Making MT Difficult

• Morphological issues with *isolating* (e.g., Vietnamese) vs. *polysynthetic* (e.g., Eskimo), and *agglutinative* (e.g., Turkish) vs. *fusion* (e.g., Russian) languages.

• Syntactic variation between *SVO* (e.g., English), *SOV* (e.g., Persian), and *VSO* (e.g., Arabic) languages.
  – SVO languages use prepositions
  – SOV languages use postpositions

• *Pro-drop* languages regularly omit subjects that must be inferred.
Lexical Gaps

• Some words in one language do not have a corresponding term in the other.
  • Rivière (river that flows into ocean) and fleuve (river that does not flow into ocean) in French
  • Schedenfraude (feeling good about another’s pain) in German.
  • Oyakoko (filial piety: devoting to parents) in Japanese
Lexical Gaps

Relation between English leg, foot, and paw, to the French jambe, pied, patte, and etape
Vauquois Triangle

Interlingua

Semantic structure

Conceptual Analysis

Semantic Transfer

Syntactic structure

Semantic Transfer

Syntactic structure

Conceptual Generation

Syntactic structure

Semantic Generation

Syntactic structure

Semantic Generation

Syntactic structure

Semantic Generation

Words

Direct translation

Morphological Analysis

Target Language

Morphological Generation

Source Language

parsing

Syntactic Transfer

Semantic Analysis
Direct Transfer

• Morphological Analysis
  – Mary didn’t slap the green witch. →
    Mary DO-PAST not slap the green witch.

• Lexical Transfer
  – Mary DO-PAST not slap the green witch.
  – Maria PAST no dar una bofetada a la verde bruja.

• Lexical Reordering
  – Maria no dar:PAST una bofetada a la bruja verde.

• Morphological generation
  – Maria no dió una bofetada a la bruja verde.
Syntactic Transfer

• Simple lexical reordering does not adequately handle more dramatic reordering such as that required to translate from an SVO to an SOV language.

• Need syntactic transfer rules that map parse tree for one language into one for another.
  - **English to Spanish:**
    • NP $\rightarrow$ Adj Noun $\Rightarrow$ NP $\rightarrow$ Noun Adj
  - **English to Japanese:**
    • VP $\rightarrow$ V NP $\Rightarrow$ VP $\rightarrow$ NP V
    • PP $\rightarrow$ P NP $\Rightarrow$ PP $\rightarrow$ NP P
Syntactic transformation from English (SVO) to Japanese (SOV)
A further sketch of the transfer approach
Semantic Transfer

• Some transfer requires semantic information.
• Semantic roles can determine how to properly express information in another language.
• In Chinese, PPs that express a goal, destination, or benefactor occur before the verb but those expressing a recipient occur after the verb.
• Transfer Rule
  – English to Chinese
    • VP → V PP[+Goal] ⇒ VP → PP[+Goal] V
Chinese Goal PPs often occurs preverbally

Jackie Chan went to Hong Kong
Interlingua representation of

*Mary did not slap the green witch.*
Statistical MT

- Manually encoding comprehensive bilingual lexicons and transfer rules is difficult.
- SMT acquires knowledge needed for translation from a *parallel corpus* that contains the same set of documents in two languages.
- The Canadian parliamentary proceedings in French and English is a well-known parallel corpus.
- First align the sentences in the corpus based on simple methods.
Picking a Good Translation

• A good translation should be **faithful** and correctly convey the information and tone of the original source sentence.

• A good translation should also be **fluent**, grammatically well structured and readable in the target language.

• Final objective:

\[
T_{best} = \arg\max_{T \in \text{Target}} \text{faithfulness}(T, S) \cdot \text{fluency}(T)
\]
Noisy Channel Model

- **GENERATIVE DIRECTION**
  - "CHANNEL SOURCE E"
    - E: Mary did not slap the green witch
  - BEST TARGET LANGUAGE SENTENCE E

- **NOISY CHANNEL**
  - P(F|E)

- **CHANNEL OUTPUT F**
  - F: María no dio una bofetada a la bruja verde

- **DECODING DIRECTION**
  - ARGMAX
    - LANGUAGE MODEL P(E) x TRANSLATION MODEL P(F|E)
Bayesian Analysis of Noisy Channel

\[ \hat{E} = \arg \max_{E \in \text{English}} P(E \mid F) \]

\[ = \arg \max_{E \in \text{English}} \frac{P(F \mid E)P(E)}{P(F)} \]

\[ = \arg \max_{E \in \text{English}} P(F \mid E)P(E) \]

Translation Model    Language Model

A decoder determines the most probable translation \( \hat{E} \) given \( F \)
Language Model

- Use a standard $n$-gram language model for $P(E)$.
- Can be trained on a large, unsupervised monolingual corpus for the target language $E$.
- Could use a more sophisticated PCFG language model to capture long-distance dependencies.
Translation Probabilities

• Assuming a **phrase aligned** parallel corpus is available or constructed that shows matching between phrases in $E$ and $F$.

• Then compute (MLE) estimate of $\phi$ based on simple frequency counts.

\[
\phi(\vec{f}, \vec{e}) = \frac{\text{count}(\vec{f}, \vec{e})}{\sum_{\vec{f}} \text{count}(\vec{f}, \vec{e})}
\]
A Tool for Creating Language Models

Downloading and Building SRILM

Building SRILM

SRILM is available only in source form. We cannot offer precompiled binaries, so you will have to download the source code and build and install the software yourself. Instructions can be found in the INSTALL file in the top-level distribution directory.

If you have previously downloaded SRILM you may want to check for the current release version and the change log.

Prerequisites

Besides the SRILM distribution, you will also need the following freely available tools:

- A template-capable ANSI-C/C++ compiler, preferably GCC version 3.4.3 or higher. Be sure to build or download g++ (the C++ compiler) component of GCC. Recent versions of Microsoft Visual Studio Express should also work.
- GNU make, to control compilation and installation.
- GNU gzip, required for many of the utility scripts.
- GNU tar to unpack the distribution, and to allow SRILM programs to handle ".z" and "gz" compressed datafiles (highly recommended).
- bzip2 to handle ".bz2" compressed files (optional).
- 7zzip to handle "7-zip" compressed files (optional).
- xz Utility to handle ".xz" compressed files (optional).
- The iconv library or equivalent, to handle UTF-encoded text files on Unix-based systems. (This library may already be integrated in your C library.)
- The Tcl embeddable scripting language library (only required for some of the test executables).
- The Cygwin porting layer, to build SRILM on a Microsoft Windows system. Cygwin comes with the packages above, although they may not all be installed by default. Alternatively, you can build native Windows binaries using the free MS Visual Studio 2010 Express compiler.

Noncommercial Use

Government agencies, and schools, universities, and non-profit organizations can download SRILM free of charge under SRI's "Research Community License", for use in projects that do not receive external funding other than government research grants and contracts. For other uses please inquire about commercial licensing.

Download

To download the source code, please first fill in the form below. Read the License document and indicate your agreement to comply with it by pressing the button below. This will start the download. The distribution is in gzipped tar format, so you should tell your browser to store the data in a file ending in ".tar.gz". You will need the "gunzip" and "tar" utilities to unpack the downloaded file into a directory structure.
A Tool for Word Alignment

**GIZA++: Training of statistical translation models.**

GIZA++ is an extension of the program GIZA (part of the SMT toolkit EGYPT) which was developed by the Statistical Machine Trans includes a lot of additional features. The extensions of GIZA++ were designed and written by Franz Josef Och.

**About GIZA++**

The program includes the following extensions to GIZA:

- Model 4;
- Model 5;
- Alignment models depending on word classes (software for producing word classes can be downloaded [here](http://www.statmt.org/));
- Implements the HMM alignment model: Baum-Welch training, Forward-Backward algorithm, empty word, dependency on word
- Includes a variant of Model 3 and Model 4 which allow the training of the parameter $p_0$;
- Various smoothing techniques for fertility, distortion, alignment parameters;
- Significant more efficient training of the fertility models;
- Correct implementation of pegging as described in (Brown et al. 1993), a series of heuristics in order to make pegging sufficient;
- ...

**In order to compile GIZA++ you may need:**

- a recent version of the GNU compiler (2.95 or higher)
- a recent version of assembler and linker which do not have restrictions with respect to the length of symbol names

It is known to compile on Linux, Irix and SUNOS systems. A lot of older compiler version do not fully support all features of STL that program. If any compilation problem occurs, please first try to get the newest compiler version. Patches to the code are most welcome.

It is released under the GNU Public License (GPL).

**Citation:**

You are welcome to use the code under the terms of the licence for research or commercial purposes, however please acknowledge its use


Here is a BibTeX entry:

```latex
@ARTICLE{och03ssc, 
AUTHOR = {Franz Josef Och and Hermann Ney}, 
TITLE = {A Systematic Comparison of Various Statistical Alignment Models}, 
JOURNAL = {Computational Linguistics}, 
NUMBER = 1, 
VOLUME = 25, 
YEAR = 2003, 
PAGES = {19--51}}
```

**Versions:**

- newest version on code.google.com NEW
- GIZA++ 2003-09-30.tar.gz
Welcome to Moses!

Moses is a statistical machine translation system that allows you to automatically train translation models algorithm quickly finds the highest probability translation among the exponential number of choices.

News

- 5 October 2017 Moses v 4.0 has been released!
- 8 September 2016 Moses2, a fast drop-in replacement for the Moses decoder
- 12 December 2015 Add a new feature function to Moses
- 17 June 2015 Slite for Windows
- 15 June 2015 Moses, and more, on Amazon cloud Box
- 1 June 2015 Developing Moses with Eclipse video
- 4 February 2015 Moses v 3.0 has been released!
- 21 July 2014 Moses now has nightly speed tests
- 14 July 2014 How to compile Moses with Eclipse
- 4 March 2014 Bug fix release for Moses, now version 2.1.1
- 3 February The 2014 Machine Translation Marathon will take place in Trento, Italy from 8-13th Sept
- 21 January 2014 Moses v 2.1 has been released!
- 26 March 2013 The 2013 Machine Translation Marathon (MTM2013) will take place in Prague, Czecl
- 5 March 2013 What do you want to see in Moses v2.0? See here for projects and how to suggest them.
- 28 January 2013 Moses v 1.0 has been released!
- 12 October 2012 Moses v 0.91 released
- February 2012: Moses development is being supported by the EU under the MosesCore project
- September 2011: Moses now has a cruise control page to see the status of the current builds
- September 2011: Moses is now hosted on github

Features

- Moses offers two types of translation models: phrase-based and tree-based
- Moses features factored translation models, which enable the integration linguistic and other informat
- Moses allows the decoding of confusion networks and word lattices, enabling easy integration with ar
- The Experiment Management System makes using Moses much easier

Get started

The released software includes a command line executable which can used for decoding. The source code for th

Learn about the decoder, training models, and tuning. Follow the step-by-step guide to build a baseline transla

Acknowledgement

The development of Moses is mainly supported by the European Union under the following projects:

- EuroMatrix and TC-STAR (Framework 6)
- EuroMatrixPlus, LetsMT, META-MET5, MosesCore and MateCat (Framework 7)

It has received additional support from

- University of Edinburgh, Scotland
- Charles University, Prague, Czech Republic
Evaluating MT

- Human subjective evaluation is the best but is time-consuming and expensive.
- Automated evaluation comparing the output to multiple human reference translations is cheaper and correlates with human judgments.
Computer-Aided Translation Evaluation

• **Edit cost**: Measure the number of changes that a human translator must make to correct the MT output.
  - Number of words changed
  - Amount of time taken to edit
  - Number of keystrokes needed to edit
Automatic Evaluation of MT

- Collect one or more human reference translations of the source.
- Compare MT output to these reference translations.
- Score result based on similarity to the reference translations.
  - BLEU
  - NIST
  - TER
  - METEOR
• Determine number of $n$-grams of various sizes that the MT output shares with the reference translations.

• Compute a modified precision measure of the $n$-grams in MT result.
BLEU Example

Cand 1: Mary no slap the witch green
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not slap the green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.

Cand 1 Unigram Precision: 5/6
Cand 1: Mary no slap the witch green.
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not slap the green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.

Cand 1 Bigram Precision: 1/5
BLEU Example

Cand 1: Mary no slap the witch green.
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not slap the green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.

Clip match count of each $n$-gram to maximum count of the $n$-gram in any single reference translation

Cand 2 Unigram Precision: 7/10
Cand 1: Mary no slap the witch green.
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not slap the green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.

Cand 2 Bigram Precision: 4/9
Modified $N$-Gram Precision

- Average $n$-gram precision over all $n$-grams up to size $N$ (typically 4) using geometric mean.

$$p_n = \frac{\sum_{C \in \text{corpus}} \sum_{n-\text{gram} \in C} \text{count}_{\text{clip}} (n - \text{gram})}{\sum_{C \in \text{corpus}} \sum_{n-\text{gram} \in C} \text{count} (n - \text{gram})}$$

$$p = \sqrt[N]{\prod_{n=1}^{N} p_n}$$

Cand 1:  $p = \frac{\sqrt{5 \times 1}}{6 \times 5} = 0.408$

Cand 2:  $p = \frac{\sqrt{7 \times 4}}{10 \times 9} = 0.558$
Brevity Penalty

• Not easy to compute recall to complement precision since there are multiple alternative gold-standard references and don’t need to match all of them.

• Instead, use a penalty for translations that are shorter than the reference translations.

• Define effective reference length, $r$, for each sentence as the length of the reference sentence with the largest number of $n$-gram matches. Let $c$ be the candidate sentence length.

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$
• Final BLEU Score: $\text{BLEU} = BP \times \rho$

  **Cand 1:** Mary no slap the witch green.
  **Best Ref:** Mary did not slap the green witch.

  \[
  c = 6, \quad r = 7, \quad BP = e^{(1-7/6)} = 0.846 \\
  \text{BLEU} = 0.846 \times 0.408 = 0.345
  \]

  **Cand 2:** Mary did not give a smack to a green witch.
  **Best Ref:** Mary did not smack the green witch.

  \[
  c = 10, \quad r = 7, \quad BP = 1 \\
  \text{BLEU} = 1 \times 0.558 = 0.558
  \]