COLX-531: Neural Machine Translation

Muhammad Abdul-Mageed

muhammad.mageed@ubc.ca

Deep Learning & NLP Lab

The University of British Columbia

Table of Contents

Word-Level Translation

Road Map

- Translate a sentence by naively translating its words (**generative modeling**)
- Assume **bilingual dictionary**: Given a word in a **foreign language** (f), what are possible **English** glosses (e)?
- **Problem:** Each foreign word can have multiple English equivalents. **Solution:** Choose the most likely based on ***parallel corpus*** stats
- **Other Problems:**
  - How to **order** words in target (English)? (Note: Some words should move together)
  - One word can translate into many words (**one-to-many**)
  - Many words can translate into only one word (**many-to-one**)
  - **Null-to-one/One-to-null**
  - ...

IBM Model 1: Lexical Translation

IBM Model 1

- **Dictionary look-up:** Haus — house, building, home, household, shell
- **Multiple translations**
  - some *more frequent* than others
  - e.g., house, and building most common
  - **special cases:** Haus of a snail is its shell

Collect Stats (Parallel Corpus) & Estimate Trans Probs (MLE)

Translation of *Haus*	Count
house	8,000
building	1,600
home	200
household	150
shell	50

$$p_f(e) = \begin{cases} 0.8 & \text{if } e = \text{house}, \\ 0.16 & \text{if } e = \text{building}, \\ 0.02 & \text{if } e = \text{home}, \\ 0.015 & \text{if } e = \text{household}, \\ 0.005 & \text{if } e = \text{shell}. \end{cases}$$

Alignment & Reordering

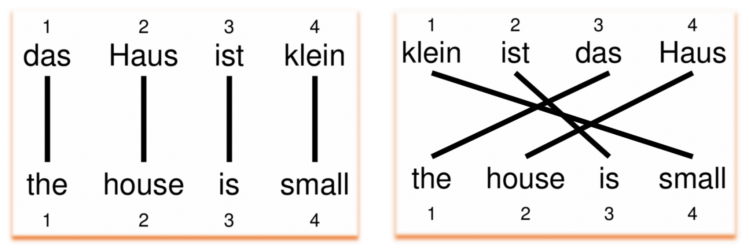


Figure: We **align** words in one language with the words in the other (**left**). For translation, words may be **reordered** using an **alignment function** α (**right**).

Word Dropping & Inserting

Dropping & Inserting

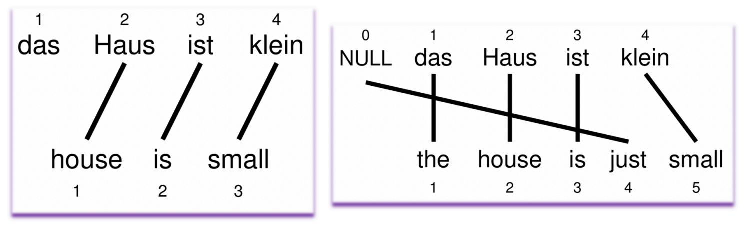


Figure: Words may be **dropped** during translation (German article das is dropped) (**left**), or **inserted** (English just does not have an equivalent in German, and so we map it to a *NULL* token)

IBM Model 1

IBM Model 1

- Generative model: break up translation process into smaller steps
  - IBM Model 1 only uses lexical translation
- Translation probability
  - for a foreign sentence **f** $= (f_1, ..., f_{l_f})$ of length l_f
  - to an English sentence **e** $= (e_1, ..., e_{l_e})$ of length l_e
  - with an alignment of each English word e_j to a foreign word f_i according to the alignment function $a : j \to i$

$$p(\mathbf{e}, a|\mathbf{f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

  - parameter ϵ is a normalization constant

Figure: (From Philipp Kohen)

IBM Model 1 *Contd.*

Example

das

e	$t(e\|f)$
the	0.7
that	0.15
which	0.075
who	0.05
this	0.025

Haus

e	$t(e\|f)$
house	0.8
building	0.16
home	0.02
household	0.015
shell	0.005

ist

e	$t(e\|f)$
is	0.8
’s	0.16
exists	0.02
has	0.015
are	0.005

klein

e	$t(e\|f)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

$$
\begin{aligned}
p(e,a|f) &= \frac{\epsilon}{4^3} \times t(\text{the}|\text{das}) \times t(\text{house}|\text{Haus}) \times t(\text{is}|\text{ist}) \times t(\text{small}|\text{klein}) \\
&= \frac{\epsilon}{4^3} \times 0.7 \times 0.8 \times 0.8 \times 0.4 \\
&= 0.0028\epsilon
\end{aligned}
$$

Figure: (From Philipp Kohen)

IBM Models 1-5

IBM Models

- Designed for **word-level translation**

Model	**Function**
IBM Model 1	lexical translation
IBM Model 2	adds absolute reordering model
IBM Model 3	adds fertility model
IBM Model 4	relative reordering model
IBM Model 5	fixes deficiency

Noisy Channel

Noisy Channel

- What is noisy channel? Watch (this).
- Originates in **acoustics** and **information theory**

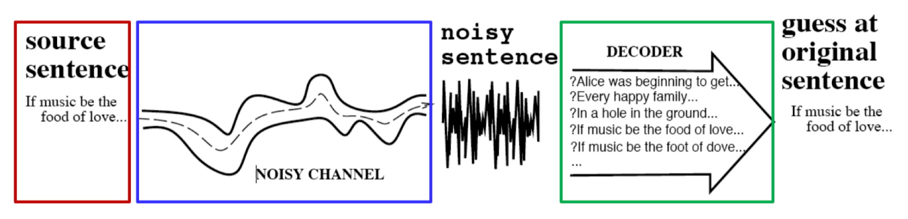
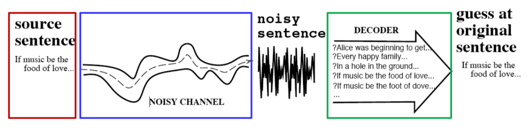


Figure: Noisy channel. (From Dan Klein)

Noisy Channel *Contd.*

Noisy Channel: Two Models

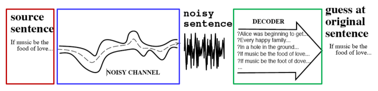


- Assume foreign sentence (message) was English (where we have a LM), but **distorted in noisy channel**. $P(\text{Source}) = P(\text{LM})$
- **Goal: Restore message** (in Eng). $P(\text{Received}|\text{Source}) = P(e|f)$

Speech	**MT**
Source	Target (Eng) **P(LM)**
Noisy Channel Model	SMT Model
Receiver (Distorted Message)	Input (foreign sent) **P(e|f)**

Noisy Channel *Contd.*

Noisy Channel: Two Models



- Use Bayes' rule to decompose **P(e|f)** into:
  - **Translation Model: P(f|e) * P(e)**
  - **Language Model: P(e)**

1: New Model

$$\text{argmax}\, P(e|f) = \text{argmax}\, \frac{P(f|e) * P(e)}{P(e)} = \text{argmax}\, P(f|e) * P(e).$$

Updated MT Model

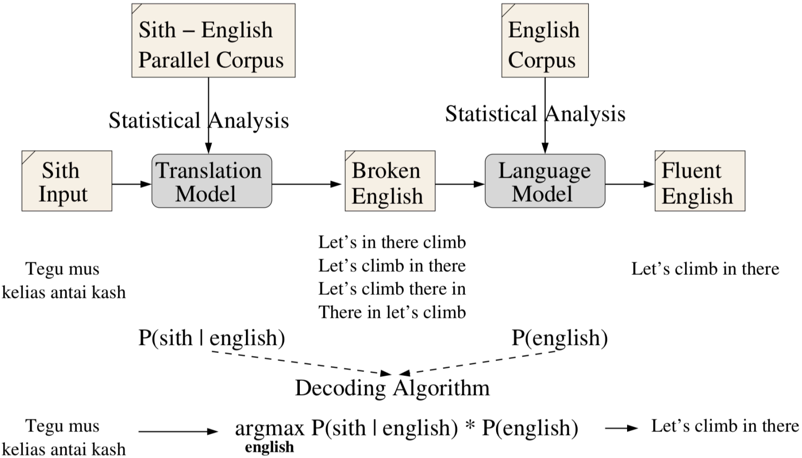


Figure: (From Fabienne Cap)

Updated MT Model *Contd.*

Translation Model: prefers **adequate** translations

- P(Tegu mus kelias antai kash|Let's climb in there) >
- P(Tegu mus kellias antai kash|Let's climb in **here**) >
- P(Tegu mus kelias antai kash|Let's **clamber** in there)

Language Model: prefers grammatical/**fluent** sequences

- P(Let's climb in there) > P(Let's there climb in)

Figure: (From Fabienne Cap)

Continuous Word Representations **Across Languages**

Background

- **Distributional hypothesis (Harris, 1954):** Words occurring in similar contexts tend to have similar meanings
- Exploited in **Word2vec** (Mikolov et al., 2013c;a) and **GloVe** (Pennington et al., 2014); and **FastText** (Bojanowski et al., 2017)
- **Exciting discovery!**: Continuous word embedding spaces **exhibit similar structures across languages**, even when considering distant language pairs like English and Vietnamese (Mikolov et al., 2013b)
- Mikolov et al. (2013b) use a **linear mapping from a source to a target embedding space** with a parallel vocabulary of 5K words as anchor points to learn this mapping
- Mikolov et al. (2013b) **evaluate on a word translation task**

Supervised Learning of XL Word Embeddings

Studies Relying on Bilingual Word Lexica

- Faruqui & Dyer (2014); Xing et al. (2015); Lazaridou et al. (2015); Ammar et al. (2016); Artetxe et al. (2016); Smith et al. (2017)

Reducing Reliance on Bilingual Lexica

- **Using identical character strings** to form a parallel vocabulary (Smith et al., 2017)
- **Using aligned digits** to gradually align embedding spaces (Artetxe et al.,2017)
- Mostly limited to **similar languages sharing a common alphabet**, such as European languages.

Unsupervised, But Less Successful!

Unsupervised

- **Using a distribution-based approach** (Cao et al., 2016)
- **Using adversarial training** (Zhang et al., 2017b)
- Both are **less successful than supervised methods**
- **Conneau et al., 2018**: **(On par with supervised methods!)**
  1. **adversarial training**
  2. **synthetic parallel vocabulary**
  3. **cross-domain similarity local scaling (CSLS)**
- With **two sets of embeddings trained independently on monolingual data**
- **Learn a mapping between the two sets such that translations are close in the shared space**

Learning a Mapping W Between S & T

Finding in Mikolov et al. (2013b)

- Let $\mathcal{X} = \{x_1, ..., x_n\}$ and $\mathcal{Y} = \{y_1, ..., y_m\}$ be **two sets of** n **and** m **word embeddings** coming from a source and a target language
- We can **exploit similarities of monolingual embedding spaces to learn a mapping** W **between source and target space**.
- P.S. They **use a dict of** $n = 5000$ **pairs of words** $\{x_i, y_i\}_{i \in \{1,n\}}$ to learn a linear mapping such that: (see next slide)

2: Main Loss

$$W^\star = \underset{W \in M_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_{\mathrm{F}}$$

Where:

- d: dimension of the embeddings
- $M_d(\mathbb{R})$: space of $d \times d$ matrices of real numbers
- X and Y: aligned matrices of $d \times n$ with embeddings of the words in parallel vocab
- **Translation** t **of any source word** s defined as: $t = \operatorname{argmax}_t \cos(Wx_s, y_t)$.

Conneau et al. (2018)

Published as a conference paper at ICLR 2018

WORD TRANSLATION WITHOUT PARALLEL DATA

Alexis Conneau[∗†‡], **Guillaume Lample**[∗†§],
Marc'Aurelio Ranzato[†], **Ludovic Denoyer**[§], **Hervé Jégou**[†]
{aconneau,glample,ranzato,rvj}@fb.com
ludovic.denoyer@upmc.fr

ABSTRACT

State-of-the-art methods for learning cross-lingual word embeddings have relied on bilingual dictionaries or parallel corpora. Recent studies showed that the need for parallel data supervision can be alleviated with character-level information. While these methods showed encouraging results, they are not on par with their supervised counterparts and are limited to pairs of languages sharing a common alphabet. In this work, we show that we can build a bilingual dictionary between two languages without using any parallel corpora, by aligning monolingual word embedding spaces in an unsupervised way. Without using any character information, our model even outperforms existing supervised methods on cross-lingual tasks for some language pairs. Our experiments demonstrate that our method works very well also for distant language pairs, like English-Russian or English-Chinese. We finally describe experiments on the English-Esperanto low-resource language pair, on which there only exists a limited amount of parallel data, to show the potential impact of our method in fully unsupervised machine translation. Our code, embeddings and dictionaries are publicly available[1].

Conneau et al. (2018) *Contd.*

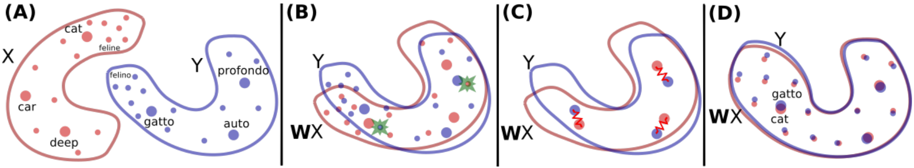


Figure: **(A)** English words in red denoted by X and Italian words in blue denoted by Y. Size of each word dot represents freq in train. **(B)** Use adversarial learning to learn a rotation matrix W which roughly aligns the two distributions. The green stars are randomly selected words fed to the discriminator to determine whether their embeddings come from the same distribution. **(C)** The mapping W is further refined via Procrustes method. **(D)** Finally, translate by using the mapping W and a distance metric that expands the space where there is high density of points (like the area around the word "cat"), so that "hubs" (like the word "cat") become less close to other word vectors than they would otherwise (compare to the same region in panel A).

Domain-adversarial Approach

Learning a Mapping W Between S & T Space

- They use **Deep Adversarial Networks**
- **Discriminator:** Trained to discriminate between elements randomly sampled from $W\mathcal{X} = \{Wx_1, ..., Wx_n\}$ and $\mathcal{Y}$.
- **Mapping W:** W trained to prevent the discriminator from making accurate predictions (Recall Generator)
- **A two-player game:** Discriminator aims at maximizing its ability to identify the origin of an embedding, and W aims at preventing the discriminator from doing so by making $W\mathcal{X}$ and $\mathcal{Y}$ as *similar* as possible

Discriminator Objective

They consider discriminator parameters to be θ_D, and the probability $P_{\theta_D}\left(\text{source} = 1 \middle| z\right)$ that a vector z is the mapping of a source embedding (as opposed to a target embedding) according to the discriminator.

3: Discriminator Loss

$$\mathcal{L}_D(\theta_D|W) = -\frac{1}{n}\sum_{i=1}^{n} \log P_{\theta_D}\left(\text{source} = 1 \middle| Wx_i\right) -$$

$$\frac{1}{m}\sum_{i=1}^{m} \log P_{\theta_D}\left(\text{source} = 0 \middle| y_i\right).$$

Mapping

Mapping Objective

In the unsupervised setting, W is now trained so that the discriminator is unable to accurately predict the embedding origins:

4: Mapping Loss

$$\mathcal{L}_W(W|\theta_D) = -\frac{1}{n}\sum_{i=1}^{n} \log P_{\theta_D}\left(\text{source} = 0 \middle| Wx_i\right) -$$

$$\frac{1}{m}\sum_{i=1}^{m} \log P_{\theta_D}\left(\text{source} = 1 \middle| y_i\right).$$

Refinement

Refining Mapping W

- Adversarial approach tries to align all words **irrespective of their frequencies**
- **Rare words** are updated less frequently, and occur in different contexts in each corpus. (harder to align)
- **Solution:** Use most freq words to acquire **synthetic parallel vocab** using W just learned with adversarial training
- They **retain only mutual nearest neighbors**, to ensure a high-quality dictionary
- Apply the **Procrustes algorithm** on dict and possibly repeat
- Procrustes offers a closed form solution obtained from the singular value decomposition (SVD) of YX^T (see paper)

Hubness Problem: Points tending to be nearest neighbors of many points in high-dimensional spaces

- Need to **improve comparison metric** such that the nearest neighbor of a source word, in the target language, is more likely to have as a nearest neighbor this particular source word
- **Problem: Nearest neighbors are asymmetric**: y being a K-NN of x does not imply that x is a K-NN of y.
- Some vectors, dubbed *hubs*, are with high probability nearest neighbors of many other points, while others (*anti-hubs*) are not nearest neighbors of any point.

Cross-Domain Similarity Local Scaling (CSLS)

Bi-partite Neighborhood Graph

- They consider a **bi-partite neighborhood graph** where each word of a given dictionary is connected to its K nearest neighbors in the other language.
- $\mathcal{N}_T(Wx_s)$: The neighborhood, on the bi-partite graph, associated with a mapped source word embedding Wx_s.
- All K elements of $\mathcal{N}_T(Wx_s)$ are words from the target language.
- Similarly, $\mathcal{N}_S(y_t)$ is the neighborhood associated with a word t of the target language.

Mean Similarity of Source Embedding

5: Mean similarity of source embedding x_s to its target neighborhood

$$r_T(Wx_s) = \frac{1}{K} \sum_{y_t \in \mathcal{N}_T(Wx_s)} \cos(Wx_s, y_t)$$

Likewise $r_S(y_t)$ is the mean similarity of a target word y_t to its neighborhood.

Compute Mean Similarities

- **Compute MS quantities for all source and target word vectors with their neighbors**, and use them to define a similarity measure $CSLS(.,.)$ between mapped source words and target words as: $CSLS(Wx_s, y_t) = 2\cos(Wx_s, y_t) - r_T(Wx_s) - r_S x(y_t)$

CSLS

Compute Mean Similarities

- The CSLS update **increases the similarity associated with isolated word vectors**
- Conversely, **it decreases the ones of vectors lying in dense areas**
- CSLS **significantly increases the accuracy for word translation retrieval**, while not requiring any parameter tuning