Machine Translation

Tell me what you sagst and I'll tell you ce que tu as dit.

Maximilian Müller-Eberstein

3.5.2017

Institut für Computerlinguistik

Ist künstliche Intelligenz gefährlich?
PD Dr. Ullrich Köthe

Introduction

Translate me this

Die Veranstaltung fand um 9 Uhr statt.

Translate me this

Die Veranstaltung fand um 9 Uhr statt.

The event took place at 9 o'clock.

Translate me this

Die Veranstaltung fand um 9 Uhr statt.

The event took place at 9 o'clock.

The event took place at 9 AM.

The presentation took place at 9:00.

The performance started at nine.

Translate me this

La manifestation s'est déroulé à 9 heures.

Translate me this

La manifestation s'est déroulé à 9 heures.

The event took place at 9 o'clock.

The event took place at 9 AM.

The presentation took place at 9:00.

The performance started at nine.

Translate me this

パワーレベルは９千以上です。

Translate me this

パワーレベルは９千以上です。

The power level is over 9 thousand.

Translate me this

パワーレベルは９千以上です。

The power level is over 9 thousand.

Their power level is over 9000.

Her power-level is above nine thousand.

His powerlevel is over 9,000.

Challenges

- Word Level
  - What even is a word? *“Die KI-Vorlesung”*
  - Word Sense Disambiguation in both languages *“Drop the bass!”*
  - Out-of-vocabulary words: *“’Murica!”*
- Phrase Level
  - Syntactic structures such as word order *“The cake a lie am.”*
  - Fluency of the word translations placed together *“Patience, you must have.”*
  - Semanticity of the phrase *“Green ideas sleep furiously.”*
- Document Level
  - Entities accross phrase-boundaries *“The chancellor [...]. She said [...].”*
  - Semanticity of the document *“Construction is ongoing. The airport opened in 2012.”*
  - Domain-specific training data *“#JustMTThings”*

History

- **1949** Warren Weaver publishes the Translation Memorandum[17]
  1. Word Sense Disambiguation using immediate context
  2. Translation as solving formal logic problems
  3. Usage of cryptographic methods, decoding the foreign language
  4. Universal Linguistics as bridge for translations
- **1954** IBM Georgetown-Experiment (Russian to English)
- **1960s** Soviet Union and USA pour research funding into MT
- **1966** ALPAC report[14] sees no cost-benefit which results in loss of funding

History

- **1970-1980** Commercial systems such as METEO[3] and SYSTRAN[16] thrive
- **1990s** (Re-)introduction of statistical MT by researchers at IBM[2]
- **1994** Online translators become available (AltaVista, Google Language Tools)
- **2001** DARPA starts funding MT extensively (especially for Arabic)
- **2012** Google translates 1 million books a day[9]
- **2016** Google switches to Neural Machine Translation[18]

Machine Translation Disambiguation

- Rule-based machine translation
  - Look-ups based on dictionaries containing vocabulary, syntax, morphology etc.
  - Encoding into and decoding from interlingual representations
  - Example-based approaches that infer new translations from known ones

- Statistical machine translation
  - statistical models which are optimised on gigantic parallel-corpora
  - generally speaking: $\text{argmax}_t p(t|s)$ with t as target and s as source
  - Neural machine translation is the currently favoured model

Statistical Models

IBM Models

Brown et al. (1990) - A statistical approach to machine translation[2]

- Formalization of translation as a statistical optimisation problem
- Introduction of IBM models 1-5 (explained in depth in [9])
- Increasingly complex models modelling the different challenges of MT
- More complex models were developed on top of these methods

IBM Models

source s	musique$_1$	jazz$_2$	musique$_1$
target t	jazz$_1$	music$_2$	music$_1$

- We have information on co-occurrence
- We are missing information on alignments $a(i) = j$
- We are missing translation probabilities $p_{word}(t_i|s_j)$

IBM Model 1

$$p_{sen}(t, a|s) = \prod_i p_{word}(t_i|s_{a(i)}) \tag{1}$$

- Sentences usually occur only once, so the task is divided
- Lexical word-by-word translation according to the highest probability

IBM Model 1

Expectation Maximization Algorithm

- Initialise uniform probability distribution
- Expectation Step
    - Use current distribution to match source- to target words
    - Normalise alignment probabilities
- Maximization Step
    - Use newly assigned probabilities to count occurrences
    - Estimate new model using these counts
- Do while the model has not converged

IBM Models 2-5

- IBM Model 2
  - Adds an alignment probability distribution $p_{sen}(\text{"jazz music"}) > p_{sen}(\text{"music jazz"})$
  - Expectation Maximization initialised with Model 1 probabilities
- IBM Model 3
  - Adds a fertility function *"Fernbahnhof"* $\rightarrow$ *"long distance train station"*
  - Iterating over all possibilities becomes infeasible, so sampling is used
- IBM Model 4 adds a relative alignment distribution
- IBM Model 5 fixes problems arising in Model 4

Language Models

The liquid output standing securely

- Contextual information (n-grams) learned from corpora in the target language
- Incorporate fluency information using the noisy-channel model

$$\mathrm{argmax}_t p(t|s) = \mathrm{argmax}_t \frac{p(s|t)p(t)}{p(s)} \tag{2}$$

- Up to 4-gram models with interpolation, back-off and smoothing[8]
- Currently neural language models are state-of-the-art

Language Models

Ensuring fluid output

- Contextual information (n-grams) learned from corpora in the target language
- Incorporate fluency information using the noisy-channel model

$$\text{argmax}_t p(t|s) = \text{argmax}_t \frac{p(s|t)p(t)}{p(s)} \quad (2)$$

- Up to 4-gram models with interpolation, back-off and smoothing[8]
- Currently neural language models are state-of-the-art

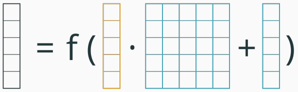
Neural Machine Translation

Neural Networks

$$\vec{y} = f(\vec{x} * W + \vec{b}) \tag{3}$$

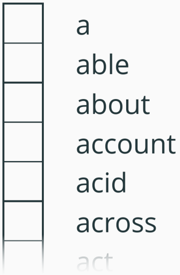
- Phrases and words must be encoded and decoded as fixed-length vectors
- Word vectors must be combined into a meaningful phrase representation
- Output must be constructed as a sequence of vectors

Neural Networks



(3)

- Phrases and words must be encoded and decoded as fixed-length vectors
- Word vectors must be combined into a meaningful phrase representation
- Output must be constructed as a sequence of vectors



"You shall know a word by the company it keeps."
- John Rupert Firth (1957)

- Corpus-based method for representing semantic meaning
- Distributional history of a word determines values in its vector
- Reduce sparsity using dimensionality reduction and smoothing
- Placement in high-dimensional space represents relations
- Word2Vec[12] (TensorFlow Embedding Projector)

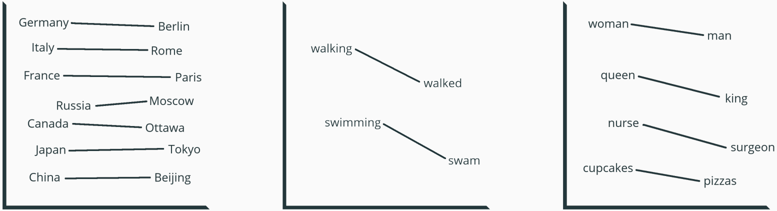
Word Embeddings

0	a
0	able
0	about
0	account
1	acid
0	across
0	act

"You shall know a word by the company it keeps."
- John Rupert Firth (1957)

- Corpus-based method for representing semantic meaning
- Distributional history of a word determines values in its vector
- Reduce sparsity using dimensionality reduction and smoothing
- Placement in high-dimensional space represents relations
- Word2Vec[12] (TensorFlow Embedding Projector)

Word Embeddings



Mikolov et al. (2013)[13] and Bolukbasi et al. (2016)[1]

- Embeddings seem to represent semantics and some syntactic features well
- Learning process carries an inherent bias depending on training data

Recurrent Neural Networks

Käsebrot ist ein gutes Brot .

- RNNs allow for multi-word sequences to be encoded in a fixed length vector
- Consideration of previous states help retain information based on word order
- Syntactic structures can also be considered during encoding
- Additional backward-pass can increase performance even further
- LSTM-[6] or GRU-cells[4] retain longer dependencies (e.g. syntactic)

Recurrent Neural Networks

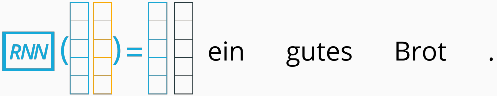
ist ein gutes Brot .

- RNNs allow for multi-word sequences to be encoded in a fixed length vector
- Consideration of previous states help retain information based on word order
- Syntactic structures can also be considered during encoding
- Additional backward-pass can increase performance even further
- LSTM-[6] or GRU-cells[4] retain longer dependencies (e.g. syntactic)

Recurrent Neural Networks

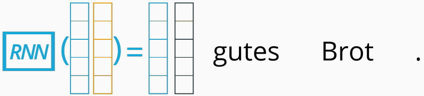
- RNNs allow for multi-word sequences to be encoded in a fixed length vector
- Consideration of previous states help retain information based on word order
- Syntactic structures can also be considered during encoding
- Additional backward-pass can increase performance even further
- LSTM-[6] or GRU-cells[4] retain longer dependencies (e.g. syntactic)

Recurrent Neural Networks



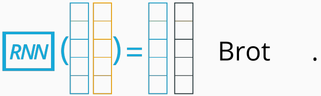
- RNNs allow for multi-word sequences to be encoded in a fixed length vector
- Consideration of previous states help retain information based on word order
- Syntactic structures can also be considered during encoding
- Additional backward-pass can increase performance even further
- LSTM-[6] or GRU-cells[4] retain longer dependencies (e.g. syntactic)

Recurrent Neural Networks



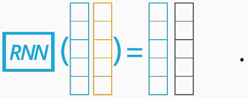
- RNNs allow for multi-word sequences to be encoded in a fixed length vector
- Consideration of previous states help retain information based on word order
- Syntactic structures can also be considered during encoding
- Additional backward-pass can increase performance even further
- LSTM-[6] or GRU-cells[4] retain longer dependencies (e.g. syntactic)

Recurrent Neural Networks



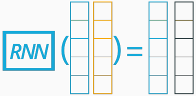
- RNNs allow for multi-word sequences to be encoded in a fixed length vector
- Consideration of previous states help retain information based on word order
- Syntactic structures can also be considered during encoding
- Additional backward-pass can increase performance even further
- LSTM-[6] or GRU-cells[4] retain longer dependencies (e.g. syntactic)

Recurrent Neural Networks



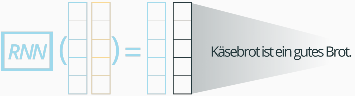
- RNNs allow for multi-word sequences to be encoded in a fixed length vector
- Consideration of previous states help retain information based on word order
- Syntactic structures can also be considered during encoding
- Additional backward-pass can increase performance even further
- LSTM-[6] or GRU-cells[4] retain longer dependencies (e.g. syntactic)

Recurrent Neural Networks



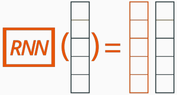
- RNNs allow for multi-word sequences to be encoded in a fixed length vector
- Consideration of previous states help retain information based on word order
- Syntactic structures can also be considered during encoding
- Additional backward-pass can increase performance even further
- LSTM-[6] or GRU-cells[4] retain longer dependencies (e.g. syntactic)

Recurrent Neural Networks



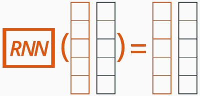
- RNNs allow for multi-word sequences to be encoded in a fixed length vector
- Consideration of previous states help retain information based on word order
- Syntactic structures can also be considered during encoding
- Additional backward-pass can increase performance even further
- LSTM-[6] or GRU-cells[4] retain longer dependencies (e.g. syntactic)

Recurrent Neural Networks



- Decoding uses the encoded source sequence to generate the target sentence
- Current word depends on previously unrolled state
- The most likely word is picked from the known target vocabulary
- Training using backpropagation through time and cross-entropy loss
- Functions similarly to a conditional language model

Recurrent Neural Networks

Cheese 

- Decoding uses the encoded source sequence to generate the target sentence
- Current word depends on previously unrolled state
- The most likely word is picked from the known target vocabulary
- Training using backpropagation through time and cross-entropy loss
- Functions similarly to a conditional language model

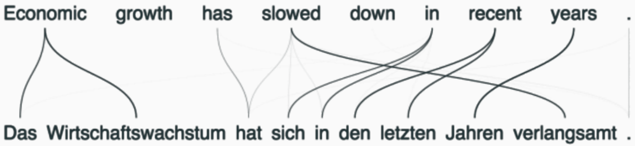
Recurrent Neural Networks

- Decoding uses the encoded source sequence to generate the target sentence
- Current word depends on previously unrolled state
- The most likely word is picked from the known target vocabulary
- Training using backpropagation through time and cross-entropy loss
- Functions similarly to a conditional language model

Cheese sandwiches are a good kind of sandwich .

- Decoding uses the encoded source sequence to generate the target sentence
- Current word depends on previously unrolled state
- The most likely word is picked from the known target vocabulary
- Training using backpropagation through time and cross-entropy loss
- Functions similarly to a conditional language model

Attention Mechanism



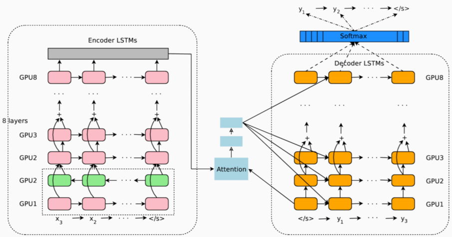
Kyunghyun Cho (2015)[1]

- It can be useful to peek at source words to translate the current target word
- Separate classifier learns relevance between input- and output states
- Relevance is treated as a probability distribution from target to source

[1] Introduction to Neural Machine Translation with GPUs - NVIDIA Devblogs

Current State

Google Neural Machine Translation



Wu et al. (2016)[18]

Since 2016, Google has been using Neural Machine Translation

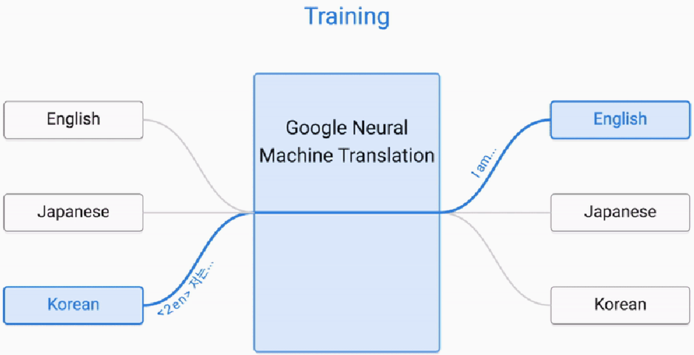
- Deep RNN with LSTM-cells in both encoder and decoder
- Embeddings of sub-word units and use of special units (e.g. numbers, word-start)
- Decoder with attention mechanism
- Reduction of translation errors by 60% and results comparable to state-of-the-art

Google Neural Machine Translation

Since 2016 (a bit later), Zero-Shot Neural Machine Translation[7] has been deployed

- Enables translation on language pairs for which there are no parallel corpora
- Target language code is prepended to the input during encoding
- Vocabulary and rest of the system are shared between languages
- Translation of multi-language phrases with different alphabets
- Semantically similar sentences are represented similarly regardless of language
- Comparable results for Fr $\rightarrow$ En and surpassing results for other language pairs

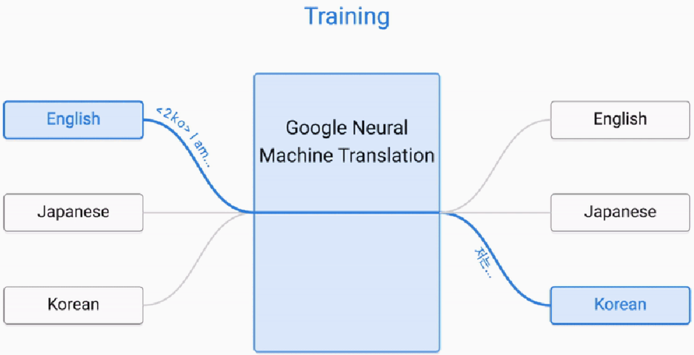
Google Neural Machine Translation



Mike Schuster et al. (2016)[2]

[2] Zero-Shot Translation with Googles Multilingual NMT System - Google Research Blog

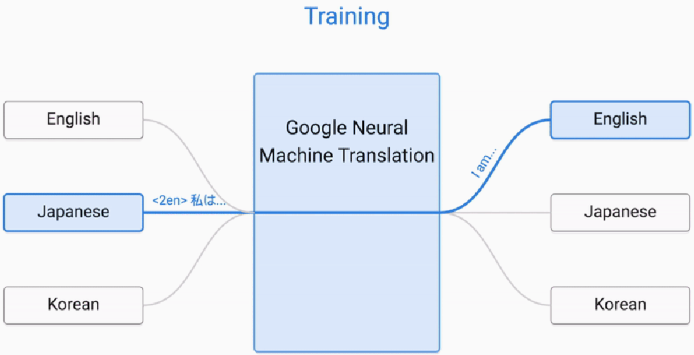
Google Neural Machine Translation



Mike Schuster et al. (2016)[2]

[2]Zero-Shot Translation with Googles Multilingual NMT System - Google Research Blog

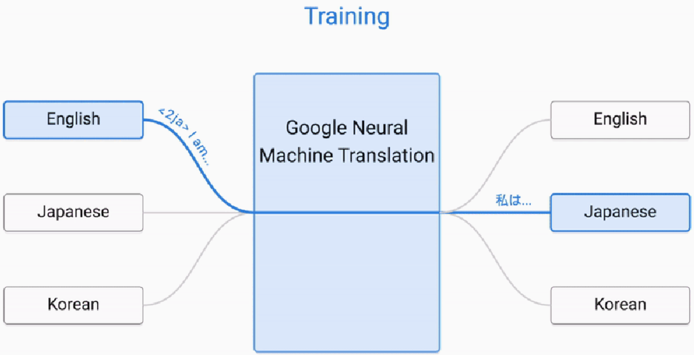
Google Neural Machine Translation



Mike Schuster et al. (2016)[2]

[2] Zero-Shot Translation with Googles Multilingual NMT System - Google Research Blog

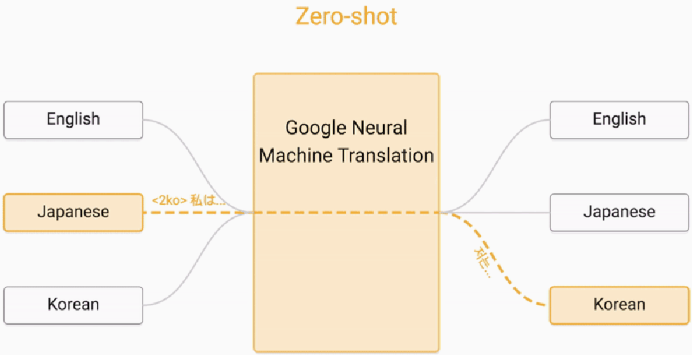
Google Neural Machine Translation



Mike Schuster et al. (2016)[2]

[2]Zero-Shot Translation with Googles Multilingual NMT System - Google Research Blog

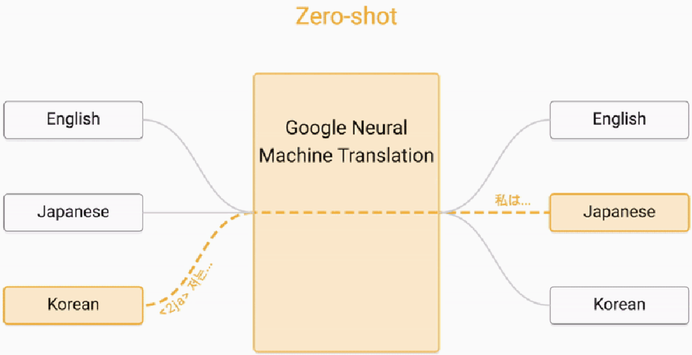
Google Neural Machine Translation



Mike Schuster et al. (2016)[2]

[2] Zero-Shot Translation with Googles Multilingual NMT System - Google Research Blog

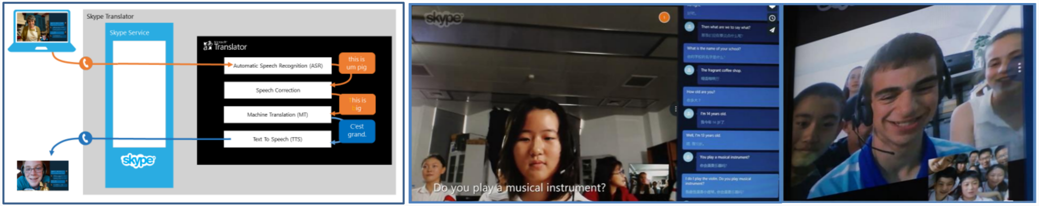
Google Neural Machine Translation



Mike Schuster et al. (2016)[2]

[2] Zero-Shot Translation with Googles Multilingual NMT System - Google Research Blog

Skype Translator



Lewis (2015)[10]

- Automated Speech Recognition
    - Challenge of recognition itself, paired with disfluency removal
    - Disambiguation of words and punctuation
- Machine Translation
    - Construction of parallel corpora for the conversational domain
    - No specifics except for statistical nature (Microsoft Translator)
- Text-to-Speech

Challenges Remaining

- The hardest word is the <UNK> you don't know
  - Copying-Mechanism: learn whether to directly copy words from source [5]
  - Byte Pair Encodings: split words into less rare subunits [15]
  - Character-Embeddings: trained on vast amounts of data [11]
- Domain-specific adaptations (e.g. medical journals, Twitter)
- Maintaining coherence over longer spans
- Metaphors, Sarcasm etc. remain difficult problems to solve

Thank you.

Questions?

References

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai.
Quantifying and reducing stereotypes in word embeddings.
arXiv preprint arXiv:1606.06121, 2016.

Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin.
A statistical approach to machine translation.
Computational linguistics, 16(2):79–85, 1990.

John Chandioux.
Meteo: an operational system, for the translation of public weather forecasts.
In *FBIS Seminar on Machine Translation. American Journal of Computational Linguistics, microfiche*, volume 46, pages 27–36, 1976.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio.
Learning phrase representations using rnn encoder-decoder for statistical machine translation.
arXiv preprint arXiv:1406.1078, 2014.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li.
Incorporating copying mechanism in sequence-to-sequence learning.
arXiv preprint arXiv:1603.06393, 2016.

Sepp Hochreiter and Jürgen Schmidhuber.
Long short-term memory.
Neural computation, 9(8):1735–1780, 1997.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean.
Google's multilingual neural machine translation system: Enabling zero-shot translation.
CoRR, abs/1611.04558, 2016.

Reinhard Kneser and Hermann Ney.
Improved backing-off for m-gram language modeling.
In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE, 1995.

References v

Philipp Koehn.
Statistical machine translation.
Cambridge University Press, 2009.

William D Lewis.
Skype translator: Breaking down language and hearing barriers.
Translating and the Computer (TC37), 2015.

Minh-Thang Luong and Christopher D Manning.
Achieving open vocabulary neural machine translation with hybrid word-character models.
arXiv preprint arXiv:1604.00788, 2016.

- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean.
  Distributed representations of words and phrases and their compositionality.
  In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig.
  Linguistic regularities in continuous space word representations.
  In *Hlt-naacl*, volume 13, pages 746–751, 2013.
- JR Pierce, JB Carroll, EP Hamp, DG Hays, CF Hockett, AG Oettinger, and A Perlis.
  Computers in translation and linguistics (alpac report). report 1416.
  National Academy of Sciences/National Research Council, 1966.

Rico Sennrich, Barry Haddow, and Alexandra Birch.
Neural machine translation of rare words with subword units.
CoRR, abs/1508.07909, 2015.

Peter Toma.
Systran as a multilingual machine translation system.
In *Proceedings of the Third European Congress on Information Systems and Networks, Overcoming the language barrier*, pages 569–581, 1977.

Warren Weaver.
Translation. memorandum. reprinted in wn locke and ad booth, eds.
Machine Translation of Languages: Fourteen Essays, 1949.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean.
Google's neural machine translation system: Bridging the gap between human and machine translation.
CoRR, abs/1609.08144, 2016.