Part V: Applications

Notes on Natural Language Processing

Chia-Ping Chen

Department of Computer Science and Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan ROC

Information Retrieval

- **Information retrieval, IR**, especially on the world-wide web, is one of the most successful applications in the digital era
  - Google
  - billions of daily searches
  - super-human
- **collection** (e.g., WWW) = source of information
- **query** = information need
- search engine: find the most relevant documents given user's query

Basic Terminology

- **document**: unit of text, *indexed*
- **collection**: a set of documents
- **term**: a lexical item that occurs in a collection
- **query**: a set of terms used by a user to express his/her information need
- **ad hoc retrieval**: an unaided user poses a query to a retrieval system, which returns relevant documents (often ordered) in a collection

Vector Space Model

- The **vector space model** for IR uses the basic idea of representing documents and queries as vectors.
  - What are the components of such a vector?
- Suppose there are n distinct terms in a collection. A document D can be represented by an n-component vector (each dimension corresponds to a distinct **term**)

$$\mathbf{d} = \begin{bmatrix} d_1 & \dots & d_n \end{bmatrix}^T .$$

- Similarly for a query Q

$$\mathbf{q} = \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix}^T .$$

Term Weight

- The component of the dimension of a term is called **term weight**.
- The **weighting scheme** refers to how weights are decided.
- The simplest weighting scheme is the **term frequency**, i.e., the number of occurrences of a term.
- The entire collection is represented by a collection of vectors, which is called a **term-by-document matrix**.

$$A = \begin{pmatrix} \uparrow & \dots & \uparrow \\ \mathbf{d_1} & \dots & \mathbf{d_N} \\ \downarrow & \dots & \downarrow \end{pmatrix}.$$

Distance

- If two vectors are similar, they are close in the vector space.
- When deciding where a query vector and a document vector is close, it is better to **normalize** the length factor.
- Therefore, the **cosine** is used, i.e.,

$$\text{sim}(\mathbf{q}, \mathbf{d}) = \frac{\sum_i q_i d_i}{|\mathbf{q}||\mathbf{d}|}.$$

- Note only the first quadrant of the vector space is used.

Document Frequency

- The **document frequency** n_i of the term w_i is the number of documents w_i occurs.
- The **inverse document frequency term weighting** is defined by

$$\text{idf}_i = \log \frac{N}{n_i}.$$

- The basic idea is that if w_i occurs only in a small number of documents ($n_i << N$), then each occurrence is important.
- Conversely, if w_i appears in many document ($n_i \sim N$), then one occurrence of w_i is not much important.

TF-IDF

- The term frequency and the inverse document frequency can be combined, called the **tf-idf weighting scheme**,

$$w_i = \text{tf}_i * \text{idf}_i$$

- For the jth document in the collection, D_j, we have

$$d_{ij} = \text{tf}_{ij} * \text{idf}_i,$$

and

$$A = \begin{pmatrix} d_{11} & \dots & d_{1N} \\ \vdots & \ddots & \vdots \\ d_{n1} & \dots & d_{nN} \end{pmatrix}.$$

Term Selection

- Some of the words are not useful for retrieval.
- We do not need to include them in the set of terms for the vector space.
- A **stoplist** is a list of high-frequency words that are excluded.
- Some words are morphological variants of the same lemma.
- It is often appropriate to treat them like they are the same.
- **Stemming** refers to such a process.

Evaluation of IR

- precision

$$p = \frac{\text{number of relevant documents retrieved}}{\text{number of documents retrieved}} = \frac{|R|}{|T|}$$

- recall

$$r = \frac{\text{number of relevant documents retrieved}}{\text{number of relevant documents in the collection}} = \frac{|R|}{|U|}$$

- When the collection is indefinite, $|U|$ is unknown, and r cannot be computed.

Mean Average Precision

- Suppose a ranked list of document is returned by an IR system.
- At the kth relevant returned document, we compute the precision *up to and including that document*.
- Specifically, suppose the kth relevant document is ranked-T_k,

$$p_k = \frac{k}{T_k}.$$

- The **mean average precision (MAP)** is then defined by

$$\text{MAP} = \frac{1}{K} \sum_{k=1}^{K} p_k.$$

Relevance Feedback

- There are some methods to improve user query.
- The most effective method in the vector space model is the use of **relevance feedback**.
- The user specifies which documents are relevant to his/her query.
- Suppose R documents are relevant and S are not. We can form a new query vector by

$$\mathbf{q}' = \mathbf{q} + \beta \left(\frac{1}{R} \sum_{i=1}^{R} \mathbf{r}_i \right) - \gamma \left(\frac{1}{S} \sum_{k=1}^{S} \mathbf{s}_k \right) .$$

Query Expansion

- A query can be **expanded** to include related terms for better retrieval.
- The terms to be added are taken from a **thesaurus**, which is essentially a list of highly correlated terms.
- A thesaurus can come from an external source.
- It can also be automatically created from the documents in a collection, such as via **term clustering**
  - In the term-by-document matrix, each row corresponds to a term.
  - Row vectors can be clustered. Each cluster is a set of correlated words.

Homonymy

- **Homonymy** is a relation that holds between words with the same forms but unrelated meanings.

  financial *bank*^{1} vs. east *bank*^{2}

- Words of the same pronunciation but different spellings are not considered homonyms (be, bee). They are called **homophones**.
- Words of the same spelling but different pronunciations are not considered homonyms (CONtent, conTENT). They are called **homographs**.
- Homonymy reduces precision of IR.

Polysemy

- **Polysemy**: multiple related meanings with a single lexeme.
- Consider blood *bank*, is this *bank* the same as *bank*^{1}?
- Polysemy refers to the related but different senses of a word, while homonymy refers to obviously different meanings.
- The difference between homonymy and polysemy can be difficult to tell.
- Polysemy also reduces IR precision.

Synonymy

- Different words with the same meaning are called **synonyms**.
- Whether two words have the same meaning can be tested by **substitutability**.
- It is hardly possible for two words to be interchangeable in all contexts.
- Synonymy reduces recall rate of IR.

Hyponymy

- A kind of relation between words is that one word is a subclass of the other.
- The more specific class is called a **hyponym**, and the more general class is called a **hypernym**. For example

car is a hyponym of *vehicle*

vehicle is a hypernym of *car*

- Hyponymy also reduces recall rate of IR.

Question Answering

- From user's perspective, a direct answer is often better than a list of documents.
- **Question answering (QA)** is the task of answering a user's question.
- If the question is regarding a fact, then it is called **factoid question answering**.
  - Particularly, if the question asked is about a **named entity**, such as a person, organization, or location.
- Figure 23.7 gives some examples.

Question Processing

- Given a question, this module extracts
  - a **query**: for input to a IR system
  - an **answer type**: specification of the kind of entity that would constitute a reasonable answer
- **Query formulation** does the first part.
- **Question classification** does the second part.

Query Formulation

- Simply use the question as query
- May need query expansion on a small collection
- Rule-based **query reformulation**

where is A $\rightarrow$ A is located in

when was laser invented $\rightarrow$ the laser was invented

Question Classification

- A given question is classified according to the expected answer type.
- A question may expect an answer of type
  - PERSON
  - CITY
  - DEFINITION
  - BIOGRAPHY
  - answer type taxonomy (Figure 23.9)
- The answer type help to focus the search.
- The answer type is an indicator for answer template.

Passage Retrieval

- A document-level retrieval is employed first to return a list of related documents in a collection.
- Next, a set of potential answer passages is extracted from the list of documents.
- The unit of passage is application-dependent
  - sections, paragraphs, sentences
  - **snippets**
- First, the answer type helps to filter out irrelevant passages.
- The remaining passages are ranked based on a small set of features.

Answer Processing

- From the retrieved passages, we want to construct an answer to the question.
- There are two classes of methods
  - **pattern extraction**: based on the expected answer type, the corresponding name entity or regular expression (pattern) is searched for
  - n**-gram tiling**: every unigram, bigram, and trigram in the snippet is weighted. These n-grams are forged into larger answers, such as through a greedy tiling algorithm.

Evaluation

- **TREC** (Text REtrieval Conference) **Q/A** track
- **mean reciprocal rank** (MRR)
  - Correct answers are known for a test question.
  - A system returns a ranked list of answers.
  - The score is the reciprocal of the rank of the first correct answer.
  - For a test set of N questions,

$$\text{MRR} = \frac{\sum_{i=1}^{N} \frac{1}{r_i}}{N}$$

Text Summarization

- **outline** of any document
- **abstract** of a scientific article
- **headline** of a news article
- **snippet** of a webpage
- **action items** of a meeting
- **summary** of an email thread
- etc.

Core Problems

- content **selection**
- information **ordering**
- sentence **realization**

Single Document Summarization

- content **selection**: choose sentences
- information **ordering**: order the sentences
- sentence **realization**: clean up the sentences

Content Selection

- **unsupervised approach**
  - Based on the log-likelihood ratio, each word can be assigned a weight of 0 or 1
  - The weight of a sentence is the sum of word weights normalized by sentence length.
  - Select those sentences with top weights.
- **supervised approach**
  - a training set of hand-created *summary extracts*
  - a classifier based on features (Figure 23.17) can be trained

Multi-Document Summarization

- The issue of **redundancy**
- **Maximal marginal relevance (MRR)**
  - the addition of a sentence s to the summary is penalized by the relevance of s and the current summary
- **coherence** and **coreference**

Summarization and QA

- Instead of a short phrase to a factoid question, users are often more interested in summarization which is more informative.
- When documents are summarized to answer a user's information need, it is called **focused summarization**.
- QA and summarization systems are combined for this task.

Summarization Evaluation

- **ROUGE**: Recall-Oriented Understudy for Gisting Evaluation

$$\text{ROUGE}-n = \frac{\text{total number of } n\text{-grams overlaps}}{\text{total number of } n\text{-gram tokens in reference summaries}}$$

  - similar to BLEU (MT)
  - recall-based

Machine Translation

- Use computers to **automate** the translation from one language to another
  - news articles
  - technical documents
  - minutes
  - restricted domain (sublanguage)
- Machine translation for *literature* can be difficult, e.g., *The Story of the Stone*, Figure 25.1.

Translation Example

- There are 4 pairs of sentences in Figure 25.1.
- The Chinese words have been replaced by the English **glosses**.
- A line between a gloss ("a Chinese word in English") and a word in the English sentence denotes their correspondence.
  - For a pair of sentences, the entire correspondence between words is called its **alignment**.

Divergence

- In these examples, there are quite a few English words that do not have any correspondence in the Chinese side.
  - determiners
  - pronouns
- Another major difference is the **word ordering**.
- The difference between languages
  - morphological
  - syntactic
  - semantic
  - lexical

Vauquois Triangle

- Figure 25.3
- **analysis**, **transfer**, and **generation**
  - **words**: direct translation
  - **syntactic**: syntactic transformation
  - **semantic**: semantic transfer
  - **interlingua**: using **meaning**

Direct Translation

- morphological analysis
- lexical transfer
- *local* reordering
  - the reordering within a phrase
- morphological generation
- Figure 25.6 for an example of direct translation.

Syntactic and Semantic Transfer

- Adj Noun vs. Noun Adj
- PP V vs. V PP
- P NP vs. NP P
- SOV vs. SVO
- etc.

Interlingua

- semantic analyzer
- meaning representation
- natural language generation

Noisy Channel Model

- Without loss of generality, suppose the translation is from French to English.
- Consider a noisy channel where the channel input is an English sentence $\mathbf{e}_c$ and the output is a French sentence $\mathbf{f}$.
- From output $\mathbf{f}$ we want to decide the optimal input $\mathbf{e}^*$.
- If the criterion is to minimizes the probability of error

$$Pr(\mathbf{e}^* \neq \mathbf{e}_c),$$

then

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} Pr(\mathbf{e}|\mathbf{f}).$$

Statistical Models

- The decoding equation can be re-written as

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} Pr(\mathbf{e}|\mathbf{f}) = \arg\max_{\mathbf{e}} \; Pr(\mathbf{e}) \; Pr(\mathbf{f}|\mathbf{e}).$$

- Therefore, we need to estimate
  - $Pr(\mathbf{e})$ through a parameterized probability $p_\theta(\mathbf{e})$, called the the **language model**
  - $Pr(\mathbf{f}|\mathbf{e})$ via a parameterized probability $p_\gamma(\mathbf{f}|\mathbf{e})$, called the **translation model**.
- In addition, we need to design a decoder for the above search problem.

Alignment

- If $\mathbf{f}$ and $\mathbf{e}$ are a pair translation sentences, there is some correspondence between the words in them.
- Such correspondence is called **alignment**.
- Specifically, if e_i and f_j are corresponded, we denote that by an **alignment variable** and draw a line between them

$$a_j = i, \quad f_j \text{ — } e_{a_j}.$$

- For those target words not aligned to any source word,

$$a_j = 0.$$

That is, $e_0 =$ NULL.

IBM Model 1

- Choose a sentence length J

$$Pr(J|\mathbf{e}) \doteq \epsilon;$$

- Choose a set of alignment values $\mathbf{a}$ according to the uniform distribution

$$Pr(\mathbf{a}|\mathbf{e}, J) \doteq \prod_{j=1}^{J} \frac{1}{I+1}, \quad I = |\mathbf{e}|;$$

- Choose the words f according to translation probability

$$Pr(\mathbf{f}|\mathbf{e}, J, \mathbf{a}) \doteq \prod_{j=1}^{J} t(f_j|e_{a_j});$$

Putting It Together

- According to IBM Model 1,

$$Pr(\mathbf{a},\mathbf{f}|\mathbf{e}) = \sum_{J'} Pr(J',\mathbf{a},\mathbf{f}|\mathbf{e}) = Pr(J,\mathbf{a},\mathbf{f}|\mathbf{e}),\ \ J = |\mathbf{f}|$$
$$= Pr(J|\mathbf{e})Pr(\mathbf{a}|\mathbf{e},J)Pr(\mathbf{f}|\mathbf{e},J,\mathbf{a})$$
$$\doteq \frac{\epsilon}{(I+1)^J}\prod_{j=1}^{J} t(f_j|e_{a_j}).$$

- Thus, the **translation model** probability is

$$Pr(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} Pr(\mathbf{a},\mathbf{f}|\mathbf{e}) \doteq \sum_{\mathbf{a}} \frac{\epsilon}{(I+1)^J}\prod_{j=1}^{J} t(f_j|e_{a_j}).$$

IBM Model 2

- In IBM Model 1, the alignment probability for f_j is a constant (uniform).

$$Pr(a_j|\mathbf{e}, J) \doteq \frac{1}{I+1}.$$

- In Model 2, this model is changed to

$$Pr(a_j = i|\mathbf{e}, J) \doteq a(i|j, J, I).$$

- It follows that

$$Pr(\mathbf{a}, \mathbf{f}|\mathbf{e}) \doteq \prod_{j=1}^{J} a(a_j|j, J, I)t(f_j|e_{a_j}).$$

Probability Factorization

- Both Model 1 and Model 2 are based on the following factorization

$$Pr(\mathbf{a}, \mathbf{f}|\mathbf{e}) = Pr(J, \mathbf{a}, \mathbf{f}|\mathbf{e}) = Pr(J|\mathbf{e})Pr(\mathbf{a}, \mathbf{f}|\mathbf{e}, J)$$

$$= Pr(J|\mathbf{e}) \prod_{j=1}^{J} Pr(a_j|a_1^{j-1}, f_1^{j-1}, \mathbf{e}, J)Pr(f_j|a_1^j, f_1^{j-1}\mathbf{e}, J)$$

- The scenario behind this factorization is to generate the French words sequentially. For a given position j
  - choose the aligned English word position a_j;
  - then choose the French word f_j;
- This is, however, not the only way to specify $Pr(\mathbf{a}, \mathbf{f}|\mathbf{e})$.

Alternative Factorization

- An alternative scenario of generating $\mathbf{f}$ is as follows.
- For each $e_i \in \mathbf{e}$, we first choose the number of words ϕ_{e_i} attached to word e_i, which is called the **fertility**.
- If the sum of fertilities is less than J, we hold e_0 responsible for the remaining words,

$$\phi_0 = J - \sum_{i=1}^{I} \phi_{e_i}.$$

- We then choose the list of words for each e_i.
- Finally, we place them in the right positions to form the sentence $\mathbf{f}$.

Notations

- $T = (T_0, \ldots, T_I)$: the lists of words for $e_0, \ldots, e_I$.
  - T is called the **tableau** of $\mathbf{e}$.
  - $T_i = \{T_{i1}, \ldots, T_{i\phi_i}\}$ is called the **tablet** of e_i, where T_{ik} is the kth word of T_i.
- $\Phi = (\phi_0, \ldots, \phi_I)$: the fertility of $e_0, \ldots, e_I$.
  - ϕ_i is the fertility of e_i.
  - Φ is a determined by T.
- Π_{ik}: the position of T_{ik} in $\mathbf{f}$.

Probability

- According to the generation process, for an instance of word lists and permutation pair (τ, π), the probability is

$$Pr(\tau, \pi|\mathbf{e}) = \prod_{i=1}^{I} Pr(\phi_i|\phi_1^{i-1}, \mathbf{e}) \times Pr(\phi_0|\phi_1^I, \mathbf{e}) \times$$
$$\prod_{i=0}^{I}\prod_{k=1}^{\phi_i} Pr(\tau_{ik}|\tau_{i1}^{k-1}, \tau_0^{i-1}, \phi_0^I, \mathbf{e}) \times$$
$$\prod_{i=1}^{I}\prod_{k=1}^{\phi_i} Pr(\pi_{ik}|\pi_{i1}^{k-1}, \pi_1^{i-1}, \tau_0^I, \phi_0^I, \mathbf{e}) \times$$
$$\prod_{k=1}^{\phi_0} Pr(\pi_{0k}|\pi_{01}^{k-1}, \pi_1^I, \tau_0^I, \phi_0^I, \mathbf{e}).$$

Alignment Probability

- Different pairs of word list and permutation may lead to the same alignment $\mathbf{a}$, so

$$Pr(\mathbf{a}, \mathbf{f}|\mathbf{e}) = \sum_{(\tau,\pi)\in<\mathbf{f},\mathbf{a}>} Pr(\tau, \pi|\mathbf{e}),$$

  where $< \mathbf{f}, \mathbf{a} >$ is the set of pairs of list and permutation to produce alignment $\mathbf{a}$ and sentence $\mathbf{f}$.

- Given $\mathbf{a}, \mathbf{f}$, the total number of *indistinguishable* pairs of permutation and word list is

$$\prod_{i=0}^{I} \phi_i!.$$

IBM Model 3 Assumptions

- $Pr(\phi_i|\phi_1^{i-1}, \mathbf{e})$ depends only on ϕ_i and e_i;
- $Pr(\tau_{ik}|\tau_{i1}^{k-1}, \tau_0^{i-1}, \phi_0^I, \mathbf{e})$ depends only on τ_{ik} and e_i;
- $Pr(\pi_{ik}|\pi_{i1}^{k-1}, \pi_1^{i-1}, \tau_0^I, \phi_0^I, \mathbf{e})$ depends on π_{ik}, i, J, and I;

Probabilities in Model 3

- **fertility probability**

$$Pr(\Phi_{e_i} = \phi | \phi_1^{i-1}, \mathbf{e}) \doteq n(\phi | e_i);$$

- **translation probability**

$$Pr(T_{ik} = f | \tau_{i1}^{k-1}, \tau_0^{i-1}, \phi_0^I, \mathbf{e}) \doteq t(f | e_i);$$

- **distortion probability**

$$Pr(\Pi_{ik} = j | \pi_{i1}^{k-1}, \pi_1^{i-1}, \tau_0^I, \phi_0^I, \mathbf{e}) \doteq d(j | i, J, I);$$

e_0

- The probability $p(\phi_0|\phi_1^I, \mathbf{e})$ is assumed to be

$$Pr(\phi_0|\phi_1^I, \mathbf{e}) \doteq \binom{\phi_1 + \phi_2 + \cdots + \phi_I}{\phi_0} p_0^{\phi_1+\phi_2+\cdots+\phi_I-\phi_0} p_1^{\phi_0}.$$

- The idea is that each word in τ_1^I independently requires an extra word with probability p_1, and no word with probability $p_0 = 1 - p_1$.

Putting Together

- Using these models, we have

$$Pr(\tau, \pi|\mathbf{e}) \doteq \prod_{i=1}^{I} n(\phi_i|e_i) \times \binom{J-\phi_0}{\phi_0} p_0^{J-2\phi_0} p_1^{\phi_0} \times \prod_{i=0}^{I} \prod_{k=1}^{\phi_i} t(\tau_{ik}|e_i) \times \prod_{i=1}^{I} \prod_{k=1}^{\phi_i} d(\pi_{ik}|i, J, I) \times \frac{1}{\phi_0!}$$

- Given $\mathbf{a}, \mathbf{f}$,

$$Pr(\mathbf{a}, \mathbf{f}|\mathbf{e}) \doteq \binom{J-\phi_0}{\phi_0} p_0^{J-2\phi_0} p_1^{\phi_0} \times \prod_{i=1}^{I} \phi_i! \times \prod_{i=1}^{I} n(\phi_i|e_i) \times \prod_{j=1}^{J} t(f_j|e_{a_j}) \times \prod_{j:a_j \neq 0}^{J} d(j|a_j, I, J).$$

Deficiency

- In Model 3, it is actually allowed that several French words occupy the same word position, i.e.,

$$\pi_{ik} = \pi_{i'k'}.$$

- These **ill-formed sentences** have non-zero probabilities, so the total probability of valid sentences is less than 1.
- This is called **deficiency**.
- Here, deficiency occurs because the assignment of the positions of new words does not rule out the positions already assigned.

Model 3 and Model 4

- The probability for assigning a position is called the **distortion probability**

$$Pr(\Pi_{ik} = j | \pi_{i1}^{k-1}, \pi_1^{i-1}, \tau_0^I, \phi_0^I, \mathbf{e}).$$

- In Model 3 it is assumed that the above probability depends only on j, i, J, I.

$$Pr(\Pi_{ik} = j | \pi_{i1}^{k-1}, \pi_1^{i-1}, \tau_0^I, \phi_0^I, \mathbf{e}) \doteq d(j|i, J, I).$$

- In Model 4, there are two probabilities for the distortion probability.

Notations

- $[i]$: the position in $\mathbf{e}$ of the ith one-word **cept**
  - a word e is a cept if $\phi_e > 0$;
  - e_0 is a one-word cept if $\phi_0 > 0$.
- $\circledast_i$: the center of the ith one-word cept
- The **head** of a cept e is the word in the list of words aligned to e, whose position in $\mathbf{f}$ is the smallest.

Displacement of Head Words

- In Model 4, there are two components for the distortion probability.
- The first is the probability of placing the **head** of a cept

$$Pr(\Pi_{[i]1} = j | \pi_1^{[i]-1}, \tau_0^I, \phi_0^I, \mathbf{e}) \doteq d_1(j - \circledast_{i-1} | A(e_{[i-1]}), B(f_j)).$$

  - $A(e), B(f)$ are word class functions.
  - $j - \circledast_{i-1}$ is called the **displacement**. It may be positive or negative.

Non-Head Words

- The second is the probability of placing the remaining words.
- For the kth word of cept i,

$$Pr(\Pi_{[i]k} = j | \pi_{[i]1}^{k-1}, \pi_1^{[i]-1}, \tau_0^I, \phi_0^I, \mathbf{e}) \doteq d_{>1}(j - \pi_{[i]k-1} | B(f_j)).$$

  - We require that $\pi_{[i]k} > \pi_{[i]k-1}$. That is, subsequent words from $\tau_{[i]}$ has to be in order.

Deficiency in Model 4

- Model 4 is still deficient.
  - Words are allowed to pile up;
  - They are also allowed to occupy before the first position and beyond the last position.
- After the placement of $\tau_1^{[i]-1}$ and $\tau_{[i]1}^{k-1}$, some vacancy positions are there for the next word $\tau_{[i]k}$.
- We can enforce the constraint that $\tau_{[i]k}$ must occupy a vacancy.
- Let $v_j = v(j, \tau_1^{[i]-1}, \tau_{[i]1}^{k-1})$ be the number of vacancies up to and including position j just before $\tau_{[i]k}$ is placed.

Model 5

- For a head word,

$$Pr(\Pi_{[i]1} = j | \pi_1^{[i]-1}, \tau_0^I, \phi_0^I, \mathbf{e})$$
$$\doteq d_1(v_j | B(f_j), v_{\circledast_{i-1}}, v_I - \phi_{[i]} + 1)(1 - \delta(v_j, v_{j-1})).$$

  - The last term enforces position j has to be vacant.
  - We need to make sure that enough vacancies are allocated for the subsequent $\phi_{[i]} - 1$ words.

- For non-head words,

$$Pr(\Pi_{[i]k} = j | \pi_{[i]1}^{k-1}, \pi_1^{[i]-1}, \tau_0^I, \phi_0^I, \mathbf{e})$$
$$\doteq d_{>1}(v_j - v_{\pi_{[i]k-1}} | B(f_j), v_I - v_{\pi_{[i]k-1}} - \phi_{[i]} + k)(1 - \delta(v_j, v_{j-1})).$$

HMM Alignment Models

- As an alternative to Model 2, the probability of $\mathbf{f}, \mathbf{a}$ given $\mathbf{e}$ can be written as

$$Pr(\mathbf{a}, \mathbf{f}|\mathbf{e}) = Pr(J|\mathbf{e}) Pr(f_1^J, a_1^J|\mathbf{e}) = Pr(J|\mathbf{e}) \prod_{j=1}^{J} Pr(f_j, a_j|f_1^{j-1}, a_1^{j-1}, \mathbf{e})$$

$$= Pr(J|\mathbf{e}) \prod_{j=1}^{J} Pr(a_j|f_1^{j-1}, a_1^{j-1}, \mathbf{e}) Pr(f_j|a_j, f_1^{j-1}, a_1^{j-1}, \mathbf{e})$$

- With obvious model assumptions, this is approximated by

$$Pr(\mathbf{a}, \mathbf{f}|\mathbf{e}) \doteq p(J|I) \times \prod_{j=1}^{J} p(a_j|a_{j-1}, I) t(f_j|e_{a_j}).$$

Data

- Machine translation models are trained with **parallel corpus.**
  - text collection that is available in two languages
  - a. k. a. **parallel text**, **bitext**
  - **Hansards**
- May need to do sentence segmentation, to produce sentence pairs,

$$(\mathbf{f}_s, \mathbf{e}_s), \quad s = 1, \ldots, S.$$

EM

- Given the alignment of each pair, we can estimate the translation probability.
- Given the translation probability, we can estimate the alignment probability
- Such a chicken-egg problem can be solved by EM.
  - In the **E**-step, we compute the expected counts for estimating the translation probability $t(f|e)$.
  - In the **M**-step, we compute the maximum-likelihood estimate of $t(f|e)$.

A Simplified Version

- Consider a simplified probability

$$Pr(\mathbf{a}, \mathbf{f}|\mathbf{e}) \doteq \prod_j t(f_j|e_{a_j}).$$

- The posterior probability of the hidden variable $\mathbf{a}$ is

$$Pr(\mathbf{a}|\mathbf{e}, \mathbf{f}) = \frac{Pr(\mathbf{a}, \mathbf{f}|\mathbf{e})}{Pr(\mathbf{f}|\mathbf{e})} = \frac{Pr(\mathbf{a}, \mathbf{f}|\mathbf{e})}{\sum_{\mathbf{a}'} Pr(\mathbf{a}', \mathbf{f}|\mathbf{e})}$$

- It is instructive to follow the steps given in the text.

Log-Linear Models

- The basic problem of MT is that we are given $\mathbf{f} = f_1^J$, and we want to choose $\mathbf{e} = e_1^I$ such that

$$\hat{e}_1^I = \arg\max_{e_1^I} Pr(e_1^I | f_1^J).$$

- As an alternative to the noisy-channel model, the **posterior probability** can be modeled directly.
- In particular, the log-linear model for posterior probability has the form

$$Pr(e_1^I | f_1^J) \doteq p_{\lambda_1^M}(e_1^I | f_1^J) \propto \exp\left(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)\right).$$

Features and Parameters

- In the log-linear model, we have
  - the feature functions $h_m(e_1^I, f_1^J)$
  - the weights λ_m
- The determination of the features and the weights is the core problem in this framework.

Decoding and Learning

- **decoding problem**: search for the hypothesis with the maximum **weighted sum of features**, i.e.,

$$\hat{e}_1^I = \arg\max_{e_1^I} \sum_m \lambda_m h_m(e_1^I, f_1^J).$$

- **learning problem**: decide the weights that maximizes the **posterior class probability** of a training set,

$$\hat{\lambda}_1^M = \arg\max_{\lambda_1^M} \sum_{s=1}^{S} \log p_{\lambda_1^M}(e_1^{I_s} | f_1^{J_s}).$$

Hidden Variables

- Typically, $Pr(e_1^I|f_1^J)$ is decomposed with additional hidden variables, such as the alignment $\mathbf{a}$. We can include such hidden variables in our models, i.e.,

$$Pr(e_1^I, a_1^J|f_1^J) \doteq p_{\lambda_1^M}(e_1^I, a_1^J|f_1^J) \propto \exp\left(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, a_1^J, f_1^J)\right).$$

- For example, in the alignment template approach, the feature functions have the following form

$$h(e_1^I, f_1^J, \pi_1^K, z_1^K).$$

Phrase Extraction

- The translation probability can be learned if we have phrase-level alignments between pairs of sentences.
- Such **phrase-level** alignments can be extracted from **word-level** alignments.
- A word-level alignment can be represented by a matrix, called **alignment matrix**.
  - An example is given in Figure 25.17 and 25.18.

The Two Directions

- There are two directions of translation given a sentence pair.
- Each direction produces a different alignment matrix.
- Specifically, this is denoted by

$$A_{e \leftarrow f} = \{(a_j, j) \mid a_j > 0\};$$
$$A_{e \rightarrow f} = \{(i, b_i) \mid b_i > 0\}.$$

$A_{e \leftarrow f}$ is for the translation is from French to English.

Symmetrizing Alignments

- **intersection**: $A = A_{e \leftarrow f} \cap A_{e \rightarrow f}$.
- **extension**: alignment (i, j) is added to A if
  - $(i = a_j, j) \in (A_{e \leftarrow f} \backslash A_{e \rightarrow f})$ and $e_{a_j} = e_i$ has not an alignment in $A_{e \rightarrow f}$ (e_i is aligned to f_0);
  - $(i, j = b_i) \in (A_{e \rightarrow f} \backslash A_{e \leftarrow f})$ and $f_{b_i} = f_j$ has not an alignment in $A_{e \leftarrow f}$ (f_j is aligned to e_0);
  - (i, j) has a neighbor that is already in A, and $A \cup \{(i, j)\}$ does not contain alignment with both horizontal and vertical neighbors.

Phrases Translation Table

- Phrase pairs **consistent** with A are extracted and stored in a **phrase translation table**, along with the translation probabilities.
- A phrase pair $\left(e_{i_1}^{i_2}, f_{j_1}^{j_2}\right)$ is consistent with an alignment matrix if
  - all dots in the rows i_1 to i_2 and columns j_1 to j_2 are contained in the corresponding submatrix.

Phrase-Based Translation Model

- Break $\mathbf{e}$ and $\mathbf{f}$ into **phrases**

$$e_1^I = \tilde{e}_1^K,\ \tilde{e}_k = e_{i_{k-1}+1}, \ldots, e_{i_k}$$
$$f_1^J = \tilde{f}_1^K,\ \tilde{f}_k = f_{j_{k-1}+1}, \ldots, f_{j_k}$$

- Align $\tilde{\mathbf{e}}$ and $\tilde{\mathbf{f}}$ at the phrase level;
  - We introduce a **permutation** π_1^K for the **alignment** at the **phrase level**. That is, $\tilde{e}_k$ is aligned to $\tilde{f}_{\pi_k}$.
- The alignment between $\tilde{e}_k$ and $\tilde{f}_{\pi_k}$ introduces some probability.

Alignment Template

- An **alignment template** z is a triplet

$$z = (F_1^{J'}, E_1^{I'}, \tilde{A}),$$

where

  - there are J' French words in this template;
  - there are I' English words in this template;
  - $F_1^{J'}, E_1^{I'}$ are the word classes of these words;
  - $\tilde{A}$ is the word-level alignment between them;

- The probability

$$Pr(z|\tilde{f}) \doteq p(z|\tilde{f})$$

can be trained from parallel corpus.

Alignment Template Model

- Suppose $\tilde{e}_k$ is aligned to $\tilde{f}_{\pi_k}$ through an **alignment template** z_k**,**

$$\tilde{e}_k \overset{z_k}{\frac{\quad}{\quad}} \tilde{f}_{\pi_k}$$

- Hence, our model introduces the hidden variables of phrase-level alignments and alignment templates,

$$\pi_1^K,\ z_1^K.$$

Feature Functions

- **alignment template selection**

$$h_{\mathsf{AT}}(e_1^I, f_1^J, \pi_1^K, z_1^K) = \log \prod_{k=1}^{K} p(z_k | f_{j_{\pi_{k-1}}+1}^{j_{\pi_k}})$$

- **word selection**

$$h_{\mathsf{WRD}}(e_1^I, f_1^J, \pi_1^K, z_1^K) = \log \prod_{i=1}^{I} p_{\mathsf{WRD}}(e_i | \{f_j | (i,j) \in A\}, E_i), \quad A = A_{\pi,\mathbf{z}}$$

- **phrase alignment**

$$h_{\mathsf{AL}}(e_1^I, f_1^J, \pi_1^K, z_1^K) = \sum_{k=1}^{K+1} |j_{\pi_k - 1} - j_{\pi_{k-1}}|, \quad j_{\pi_0} = 0, j_{\pi_{K+1}-1} = J.$$

Feature Functions

- **language model features**

$$h_{\mathsf{LM}}(e_1^I, f_1^J, \pi_1^K, z_1^K) = \log \prod_{i=1}^{I+1} p_{\mathsf{LM}}(e_i | e_{i-l+1}, \ldots, e_{i-1})$$

$$h_{\mathsf{CLM}}(e_1^I, f_1^J, \pi_1^K, z_1^K) = \log \prod_{i=1}^{I+1} p_{\mathsf{CLM}}(C(e_i) | C(e_{i-m+1}), \ldots, C(e_{i-1}))$$

- **word penalty**

$$h_{\mathsf{WP}}(e_1^I, f_1^J, \pi_1^K, z_1^K) = I.$$

Search Problem

- Given f_1^J, the decision rule for optimal e_1^I is

$$\hat{e}_1^I = \arg\max_{e_1^I, \pi_1^K, z_1^K} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J, \pi_1^K, z_1^K) \right\}.$$

- For simplicity, we consider the case of using the feature functions of AT, AL, WRD, and LM (trigram),

$$\begin{aligned}\hat{e}_1^I = \arg\max_{e_1^I, \pi_1^K, z_1^K} & \sum_{i=1}^{I} [\lambda_{\mathsf{LM}} \log p_{\mathsf{LM}}(e_i|e_{i-2}, e_{i-1}) + \lambda_{\mathsf{WRD}} \log p_{\mathsf{WRD}}(e_i|\{f_j|(i,j) \in A\}, E_i)] \\ & + \sum_{k=1}^{K} \left[\lambda_{\mathsf{AT}} \log p(z_k|f_{j_{\pi_k-1}+1}^{j_{\pi_k}}) + \lambda_{\mathsf{AL}}|j_{\pi_k-1} - j_{\pi_{k-1}}| \right] \\ & + \lambda_{\mathsf{AL}}|J - j_{\pi_k}| + \lambda_{\mathsf{LM}} \log p_{\mathsf{LM}}(\mathsf{EOS}|e_{I-1}, e_I).\end{aligned}$$

Structure of Search Space

- We have grouped the contribution of feature functions into those for each word, those for each alignment template, and those for end-of-sentence.
- Accordingly, the search space is structured to take advantage of such decomposition of the objective function.
- Specifically, a search hypothesis
  - corresponds to a prefix of English sentence
  - is extended by appending one English word to generate new hypotheses

Search Graph

- The set of all hypotheses can be structured as a **graph**.
  - a node n has a hypothesis;
  - there is a directed edge from node n_1 to node n_2 if the hypothesis of n_2 is obtained by appending one word to that of n_1;
  - each edge is associated with a cost related to the feature functions;
- Let the source node be sentence start, and goal nodes be complete translations.
- The search problem is now a **graph search problem** for the path with the minimum cost.

Alignment Template Instantiation

- An **alignment template instantiation** is the application of an alignment template to a phrase.
- Given f_1^J, the set of all applicable alignment template instantiations is

$$\left\{ Z = (z, j) \mid z = (F_1^{J'}, E_1^{I'}, \tilde{A}) \ \wedge \ \exists\, j : p(z|f_j^{j+J'-1}) > 0 \right\}$$

Decision

- A **decision** is a triplet $d = (Z, e, l)$ consisting of an alignment template instantiation Z, the generated word e, and the index l of e in Z.
- A hypothesis $n = e_1^i$ corresponds to a valid sequence of decisions d_1^i.
- Any valid and complete sequence of decisions d_1^{I+1} uniquely corresponds to a translation e_1^I, a segmentation to K phrases, the phrase-level alignment π_1^K, and the alignment template instantiations z_1^K.

Possible Decisions

- Start a new alignment template: $d_i = (Z, e_i, 1)$. The incurred costs include AL and AT in addition to LM, WRD, due to a new template Z and a new word e_i.
- Extend an alignment template: $d_i = (Z, e_i, l)$. The incurred costs include just LM, WRD due to the addition of e_i.
- Finish the translation sentence: $d_i = (\mathrm{EOS}, \mathrm{EOS}, 0)$. The incurred costs include AL and LM for EOS.

Evaluation

- Human raters
- BLEU

$$\textbf{BLEU} = \text{BP} \times \exp\left(\frac{1}{n}\sum_{i=1}^{n}\log p_i\right)$$

  - p_i is the *modified* i-gram **precision**
  - BP is **brevity penalty**

$$\text{BP} = \begin{cases} 1, & c > r \\ e^{(1-r/c)}, & c \leq r \end{cases}$$

where c is the length of the candidate translation, and r is the length of the reference translation