UNIVERSIDADE DE SÃO PAULO

INSTITUTO DE CIÊNCIAS MATEMÁTICAS E CIÊNCIAS COMPUTACIONAIS

SCC0633/SCC5908 - Processamento de Linguagem Natural, 2024

- **Nome dos integrantes do grupo:**
  - Arthur Santorum Lorenzetto, NUSP: 12559465
  - Bruno Berndt Lima, NUSP: 12542550
  - Eduardo Z. Monteiro, NUSP: 12559490
  - Pedro Henrique de Freitas Maçonetto, NUSP: 12675419
  - Vinicius Kazuo Fujikawa Noguti, NUSP: 11803121

- **Tópico do trabalho:**
  - Tradução Automática.

- **Lista de córpus escolhidos para o trabalho:**

  - http://www.nilc.icmc.usp.br/nilc/tools/Fapesp%20Corpora.htm

Proposta:

1. Introdução e Motivação

Com a invenção da internet, as pessoas de hoje em dia têm acesso a conteúdos produzidos em todos os locais do mundo, mas muitas vezes pode não ser fácil entender o que se está querendo comunicar, já que pode haver uma barreira linguística entre o autor e o consumidor desses conteúdos.

Muito antes disso, os pesquisadores de PLN já buscavam um sistema capaz de fazer tradução entre duas línguas, com o objetivo de baratear e acelerar o processo de tradução, mas com a internet, hoje em dia é ainda mais interessante o desenvolvimento de um sistema do tipo, dado que ele pode ser utilizado diariamente para compreender conteúdos encontrados em línguas que o indivíduo não entende.

A princípio o tradutor proposto aqui é um tradutor da língua portuguesa para a língua inglesa, essa decisão foi tomada pois inglês é a língua que os autores têm mais familiaridade, e portanto é mais fácil avaliar as traduções geradas pelo sistema.

2. Corpus Utilizado

O corpus que será utilizado para treinamento do modelo será o “REVISTA PESQUISA FAPESP PARALLEL CORPORA”, um corpus construído a partir de textos da revista “Revista Pesquisa FAPESP”. O corpus é formado por textos alinhados para as combinações de línguas: Português-Inglês e Português-Espanhol, para o trabalho serão utilizados apenas os textos da combinação Português-Inglês.

Os textos estão organizados na forma de arquivos paralelos, de modo que existem dois arquivos com o mesmo nome com as extensões .pt e .en, que armazenam o mesmo texto nas duas línguas. Dentro dos arquivos os textos estão organizados de forma que cada linha representa uma frase, dessa forma, a linha 1 dos arquivos nome.pt e nome.en representam a mesma frase nas respectivas línguas. O formato dos arquivos facilitará bastante o trabalho, pois esse é o formato esperado pelo modelo MOSES, que será apresentado na seção 3 e utilizado no projeto, sendo assim, é esperado que o tratamento necessário seja mínimo nas bases antes do treinamento.

Uma preocupação é quanto à natureza do corpus, dada a fonte, é esperado que os textos utilizados sejam escritos em linguagem científica, o que resultará em boas traduções para frases em jargão acadêmico, mas que poderá pecar especialmente em textos de outros contextos. Adicionalmente, é esperado que o texto publicado na revista seja cuidadosamente revisado, de forma que deverá conter poucos erros, sendo assim, o tradutor final poderá ter grande dificuldade em traduzir frases com erros.

3. O Modelo Estatístico (Moses)

O funcionamento do modelo pode ser dividido nas seguintes partes.

3.1. Pré-Processamento

- **Tokenização:**
  Divisão do texto em pequenas unidades.
- **Truecasing:**
  Ajuste de capitalização das palavras.
- **Normalização:**
  Ajuste ou remoção de caracteres especiais.

3.2. Tradução

3.2.1. Alinhamento

- Extração de pares de frases utilizando um corpus alinhado**.**
- Alinhamento probabilístico utilizando GIZA++

3.2.2. Decodificação

3.2.2.1. Modelo de tradução:

Utilização da probabilidade condicional da frase de destino dada a frase de origem.

3.2.2.2. Modelo de linguagem:

Utilização de um modelo, tipicamente baseado em n-gramas, para calcular a probabilidade da sequência de palavras no idioma de destino. Garante maior fluência de acertos gramaticais do modelo final.

3.2.2.3. Combinação de modelos:

Combina os dois processos anteriores, utilizando um algoritmo de busca para encontrar a sequência de traduções mais prováveis dentre diferentes combinações.

3.3. Pós-Processamento

- **Des-tokenização**:
  Junção dos tokens gerados anteriormente, junto com tratamentos de espaços em branco caso existam.
- **De-truecasing:**
  Ajuste de capitalização das palavras.
- **Demais ajustes**:
  Remoção de espaços em branco extras; Correção de pontuação; Aplicação de regras gramaticais específicas da língua de destino; Etc.

4. Preparação dos Dados

Para preparar os dados para tradução, é necessário seguir estas etapas:

4.1. Alinhamento de Sentenças

Dividir as sentenças em dois arquivos separados, um para as sentenças em português e outro para as sentenças em inglês, garantindo que estejam alinhadas, ou seja, as linhas correspondentes de cada arquivo representam o texto e sua tradução.

4.2. Limpeza do Corpus

Remover linhas vazias e ajustar o formato das sentenças para que cada uma ocupe uma linha, sem linhas em branco entre elas.

Utilizar um script ou ferramenta como lowercase.perl para converter todas as letras para minúsculas.

4.3. Eliminação de Sentenças Longas

Remover quaisquer sentenças que excedam 100 palavras para otimizar o treinamento e melhorar a qualidade do modelo.

5. Treinamento do Modelo

Etapas de treinamento do modelo para tradução

5.1. Preparação dos Dados

Nesta fase, o corpus paralelo precisa ser convertido para um formato adequado ao toolkit GIZA++. Isso inclui a geração de dois arquivos de vocabulário e a conversão do corpus paralelo em um formato numerado.

5.2. Execução do GIZA++

O GIZA++ é uma implementação gratuita dos modelos IBM. Ele é usado para estabelecer os alinhamentos de palavras e é uma etapa inicial crucial no processo de treinamento. Os alinhamentos de

palavras são obtidos a partir da interseção de execuções bidirecionais do GIZA++, juntamente com alguns pontos de alinhamento adicionais.

5.3. Alinhamento de Palavras

Para estabelecer os alinhamentos de palavras com base nos dois alinhamentos do GIZA++, várias heurísticas podem ser aplicadas. A heurística padrão, grow-diag-final, começa com a interseção dos dois alinhamentos e depois adiciona pontos de alinhamento adicionais.

5.4. Obtenção da Tabela de Tradução Léxica

Com base nesses alinhamentos, é bastante direto estimar uma tabela de tradução léxica de máxima verossimilhança. Estimamos a tabela de tradução de palavras w(e|f) e sua inversa w(f|e).

5.5. Extração de Frases

Nesta etapa, todas as frases são agrupadas em um único arquivo.

5.6. Score das Frases

Na etapa de “scorar” as frases, as frases extraídas são avaliadas quanto à sua relevância e qualidade na tradução. Essa pontuação é baseada em diversos fatores, como a probabilidade de tradução das frases e pesos lexicais associados a cada palavra. Essas informações são usadas para criar uma tabela de tradução de frases, que pode incluir medidas como a probabilidade direta de tradução, pesos lexicais diretos e inversos, entre outros. Essas pontuações ajudam a determinar quais traduções são mais confiáveis e adequadas para o modelo de tradução.

5.7. Construção do Modelo de Reordenação Lexical

No processo de construção do modelo de reordenação, são desenvolvidos modelos que determinam a ordem das palavras durante a tradução. O modelo padrão é baseado na distância de reordenação, mas é possível construir modelos adicionais de reordenação lexicalizados. Esses modelos consideram diferentes aspectos da ordem das palavras e são especificados por uma string de configuração. Durante o treinamento, esses modelos são ajustados e configurados para melhorar a qualidade da tradução final.

5.8. Construção Modelo de Geração

O modelo de geração é construído a partir do lado alvo do corpus paralelo. Por padrão, são calculadas probabilidades para frente e para trás. Se usar o comando `--generation-type single`, apenas as probabilidades na direção do passo são calculadas.

5.9. Criação do Arquivo de Configuração

Como última etapa, um arquivo de configuração para o decodificador é gerado com todos os caminhos corretos para o modelo gerado e uma série de configurações de parâmetros padrão.

Este arquivo é chamado de `model/moses.ini`.

6. Utilização do Modelo Treinado

Uma vez que o modelo foi treinado, sua utilização é bastante simples, pode ser feita utilizando o comando:

"~/mosesdecoder/bin/moses -f ~/working/mert-work/moses.ini"

Porém como mencionado na documentação do modelo baseline, o modelo pode demorar alguns minutos para iniciar quando utilizado dessa maneira. Para evitar essa demora é recomendado binarizar a tabela de frases e a componente do modelo que faz a reordenação das palavras, isso é feito executando a seguinte sequência de comandos:

mkdir ~/working/binarised-model → Cria estrutura de diretórios

cd ~/working → Muda para o diretório

```
~/mosesdecoder/bin/processPhraseTableMin \
  -in train/model/phrase-table.gz -nscores 4 \
  -out binarised-model/phrase-table
```
→ Faz a binarização da tabela de frases.

```
~/mosesdecoder/bin/processLexicalTableMin \
  -in train/model/reordering-table.wbe-msd-bidirectional-fe.gz \
   -out binarised-model/reordering-table
```
→ Faz a binarização do modelo de ordenação das palavras.

Após isso, é necessário mudar o arquivo do modelo para a pasta "working" e mudar a referência para a tabela e modelo mencionados acima, isso é feito alterando a variável PhraseDictionaryMemory para PhraseDictionaryCompactos, e os valores das variáveis do modelo PhraseDictionary e LexicalReordering para os seguintes:

PhraseDictionary → $HOME/working/binarised-model/phrase-table.minphr

LexicalReordering → $HOME/working/binarised-model/reordering-table

Uma vez que esse processo for realizado já é possível executar o modelo com um tempo de execução bastante menor.

7. Ilustração do Funcionamento

Para tentar exemplificar o processo realizado pelo Moses, vamos fazer uma simples ilustração de como funcionam as etapas do modelo.

Para tal ilustração, iremos utilizar as seguintes frases de origem e destino:

- Frase de origem (Português): “Como você está?”
- Frase de destino (Inglês): “How are you?”

1. **Pré-processamento**
    a. Tokenização:
    Extrair os tokens da frase de origem
    Resultado = ["Como", "você", "está", "?"]
    b. Truecasing:
    Descapitalização dos tokens
    Resultado = ["como", "você", "está", "?"]
    c. Normalização:
    Remoção dos caracteres especiais
    Resultado = ["como", "voce", "esta"]

2. **Tradução**
    a. Alinhamento:
        i. Utilizamos um corpus de textos alinhados em português e inglês para extrair frases paralelas que servirão como base para a tradução.
        ii. Após a extração dos pares de frases, utilizamos o GIZA++ para calcular as probabilidades de correspondência entre as palavras "como" e "how", "você" e "are", "está" e "you", levando em consideração seu contexto no corpus bilíngue.
    b. Decodificação:
        i. Na etapa de decodificação, com as probabilidades calculadas de ocorrência condicional das palavras e as relações de probabilidade do modelo n-gramas. O software é responsável por determinar a melhor tradução considerando as probabilidades de tradução.
        ii. Como podem haver diversas traduções possíveis para tokens unitários ou compostos. O algoritmo de busca se encarrega de procurar a solução mais viável.

3. **Pós-processamento**
    a. Des-tokenização:
    Junção dos tokens em uma frase completa
    Entrada: ["How", "are", "you", ""]
    Saída: "How are you"

    b. De-truecasing:
    Recapitalização da palavra correta
    Entrada: "how are you"
    Saída: "How are you"

c. Demais ajustes
Correção de pontuação, aplicação de regras gramaticais específicas.
Entrada: "How are you"
Saída: "How are you?"

8. Referências

- http://www2.statmt.org/moses/?n=Moses.Overview
- http://www2.statmt.org/moses/?n=Moses.Baseline
- http://www2.statmt.org/moses/?n=FactoredTraining.HomePage#ntoc1
- http://www2.statmt.org/moses/?n=FactoredTraining.PrepareTraining
- http://www.nilc.icmc.usp.br/nilc/tools/Fapesp%20Corpora.htm