**北京信息科技大学**

**科研项目实践（2）**

**结题报告**

**题目： 中英机器翻译**

**学院： 计算机学院**

**专业： 计算机科学与技术**

**学生姓名： 许志豪 班级学号：计科1906 2019010093**

**学生姓名： 陈 浩 班级学号：计科1905 2019010132**

**学生姓名： 殷绍轩 班级学号：计科1905 2018010636**

**指导老师/督导老师： 王楠**

**学年/学期： 2021-2022学年第一学期**

[摘要 2](#_Toc160)

[第一章 3](#_Toc161)

[文献机器翻译现状与进展 4](#_Toc162)

[1.1 项目简介 4](#_Toc163)

[1.2 项目背景和研究现状 4](#_Toc164)

[1.3项目实施与组内分工 5](#_Toc165)

[1.3.1 项目实施 5](#_Toc166)

[1.3.2组内分工与合作 6](#_Toc167)

[第二章 7](#_Toc168)

[机器翻译模型与算法 7](#_Toc169)

[Encoder-Decoder 7](#_Toc170)

[Seq2Seq 7](#_Toc171)

[缺陷 7](#_Toc172)

[Attention 7](#_Toc173)

[Transformer 8](#_Toc174)

[SimpleNMT 8](#_Toc175)

[第三章 8](#_Toc176)

[机器翻译模型实践和文献机器翻译工程 9](#_Toc177)

[3.1机器翻译模型实践 9](#_Toc178)

[3.1.0 数据集准备 9](#_Toc179)

[3.1.1 模型训练 9](#_Toc180)

[3.2 文献机器翻译工程 10](#_Toc181)

[3.2.1建立翻译API 11](#_Toc182)

[3.2.2 后台搭建 11](#_Toc183)

[3.2.3 项目成果展示 15](#_Toc184)

[第四章 16](#_Toc185)

[项目总结与讨论 16](#_Toc186)

[4.1项目总结 16](#_Toc187)

[4.2项目收获 16](#_Toc188)

[参考文献： 18](#_Toc189)

# 摘要

使用OpenNMT、SimpleNMT等机器翻译框架训练翻译模型，先用英语-德语作为测试。测试后得到较不错结果，选择一个较为轻量级的翻译框架SimpleNMT打造本项目的机器翻译模型。不同的文化拥有独特各异的语言系统，机器没有复杂性来理解或识别俚语、行话、双关语和习语，因此，其所产生的翻译可能并不符合文化的价值观和特定规范。基于此类情况，我们严格选择了中英文数据集来源PatentGLUE。本项目使用simpleNMT框架对六万多行的中英文平行语料训练，得到一个中英文互译模型，该模型结合了学界最流行的机器翻译模型，初步实现中英文的翻译，并构建了Web端实时中英文互译翻译系统。

**关键词**： 机器翻译，SimpleNMT，实时翻译系统

# 第一章

# 文献机器翻译现状与进展

## 1.1 项目简介

机器翻译速度快捷,方便实用,并且具有极高的成本效益。它可以在几秒钟内翻译大量文本。同时,机器翻译价格低廉,有数百个免费的用户友好型应用程序,只需按下手指,便可随时随地翻译文本、图像和语音。不仅如此,一个翻译程序可以翻译数百种不同的语言。这是机器翻译的亮点,也是专业翻译人员无法实现的。

本项目使用simpleNMT框架对六万多行的中英文平行语料训练，得到一个中英文互译模型，该模型结合了学界最流行的机器翻译模型（如下），初步实现中英文的翻译，并构建了Web端实时中英文互译翻译系统。

1. [bentrevett/pytorch-seq2seq](https://gitee.com/link?target=https%3A%2F%2Fgithub.com%2Fbentrevett%2Fpytorch-seq2seq)
2. [harvardnlp/annotated-transformer](https://gitee.com/link?target=http%3A%2F%2Fnlp.seas.harvard.edu%2F2018%2F04%2F03%2Fattention.html)
3. [jadore801120/attention-is-all-you-need](https://gitee.com/link?target=https%3A%2F%2Fgithub.com%2Fjadore801120%2Fattention-is-all-you-need-pytorch)
4. [fairseq](https://gitee.com/link?target=https%3A%2F%2Fgithub.com%2Fpytorch%2Ffairseq), [OpenNMT](https://gitee.com/link?target=https%3A%2F%2Fopennmt.net%2F), [Tensor2Tensor](https://gitee.com/link?target=https%3A%2F%2Fgithub.com%2Ftensorflow%2Ftensor2tensor)

## 1.2 项目背景和研究现状

机器翻译在具有速度快、成本低优势的同时,也存在一定的缺陷。首先,机器翻译不具备文化敏感度。人类也许永远不能对机器进行编程使其理解和体验特定的文化。不同的文化拥有独特各异的语言系统,机器没有复杂性来理解或识别俚语、行话、双关语和习语,因此,其所产生的翻译可能并不符合文化的价值观和特定规范。这也是机器亟需克服的挑战之一。

其次,机器翻译无法将文字与语境义联系起来。在很多语言中,同一个单词可能会具有多个完全不相关的含义。在这种情况下,语境会对词义产生很大的影响,而对词义的理解又在很大程度上取决于从语境中读出的线索。只有人类才能将词语与语境结合起来,确定其真实含义,并创造性地对语言进行打磨,得出完整精确的译文。对机器翻译而言,这无疑是非常困难的。

第三,几乎所有的机器翻译都需要人工后期编辑。也许有一天,计算机将获得与人类相似的自然语言系统,但无论如何,后期编辑都是不可抛弃的。在机器完成翻译之后,必须有校对者对翻译进行最后的修改,以确保它使用了正确的语法,表达了可被理解的语义。

鉴于上面提到的缺点,关于机器是否会取代人工翻译的争论似乎是不必要的。在大多数情况下,这个问题的答案是否定的。虽然我们不能低估机器翻译的潜力,但我们需要明白,只要对译本的语言准确度有所要求,机器翻译就无法取代人类。相反,他们可以帮助人类完成各种庞大繁琐的任务。

近年来，自然语言处理的研究已经成为热点，而机器翻译作为自然语言研究领域的一个重要分支，同时也是人工智能领域的一个课题，同样为大家所关注。纵观机器翻译的研究历史，从上个世纪4o年代英国工程师Booth和美国工程师Weaver提出利用计算机进行翻译的想法，到5o年代欧美国家投入大量的人力、物力致力于机器翻译的研究，再到6o年代ALPAC置疑报告的提出，机器翻译走向沉寂。最近的二十年，随着语言学理论的发展、计算机技术的进步以及统计学和机器学习方法在自然语言处理领域中的广泛应用，人们对机器翻译本身的应用背景、目标等也有了更加准确的认识，机器翻译在此背景下取得了长足的发展，基于统计、基于实例等新的机器翻译方法也都是在这一时期出现，一些机器翻译系统也从实验室走向了市场。在中国，机器翻译的研究从50年代开始，多家大学和研究机构先后开发出俄汉、英汉、汉英、日汉、汉日等机器翻译系统，同时在汉语的自然语言理解方面做了大量的研究。在看到机器翻译研究取得进展的同时，我们也知道。由于对语言规律本身认识仍然不足。以及计算机对语言理解的局限性，再加上不同语言之间存在着语法结构、构造方式、语言习惯、社会背景等等的不同，机器翻译的效果与大家所期待的仍有非常大的差距。

## 1.3项目实施与组内分工

### 1.3.1 项目实施

项目自2021年9月25日正式开始。

9月25日～10月14日：机器翻译原理书籍《机器翻译：基础与模型》阅读，了解Transformer、LSTM模型等原理。

10月15日～11月 19日：使用OpenNMT、SimpleNMT等机器翻译框架训练翻译模型，先用英语-德语作为测试。测试后得到较不错结果，选择一个较为轻量级的翻译框架——SimpleNMT打造本项目的机器翻译模型。同时搭建前后端的Web实时翻译系统，先使用百度翻译API进行测试，测试结果较为满意。

11月20日～12月25日：使用中英文语料PatenGLUE作为数据集进行训练，在得到的模型中选取最优的一个作为实时翻译系统调用模型。编写Flask API来支持对翻译模型的调用，并且结合之前的写好的Web翻译框架成功部署到服务器上。至此整个项目结束。

### 1.3.2组内分工与合作

陈浩，主要负责实现数据集存储与前后端调用模型实现中英翻译。

许志豪，主要负责数据制作、模型训练以及翻译API的创建。

殷绍轩，负责收集、整理数据集、模型调研报告的撰写和幻灯片的制作。

# 第二章

# 机器翻译模型与算法

## Encoder-Decoder

将现实问题转化为数学问题，通过求解数学问题，从而解决现实问题

Encoder：编码器，将现实问题转化为数学问题

Decoder：解码器，求解数学问题，并转化为现实世界的解决方案

## Seq2Seq

所谓Seq2Seq(Sequence to Sequence), 就是一种能够根据给定的序列，通过特定的方法生成另一个序列的方法。它被提出于2014年，最早由两篇文章独立地阐述了它主要思想，分别是Google Brain团队的《Sequence to Sequence Learning with Neural Networks》和Yoshua Bengio团队的《Learning Phrase Representation using RNN Encoder-Decoder for Statistical Machine Translation》。这两篇文章不谋而合地提出了相似的解决思路，Seq2Seq由此产生。文本到文本，为了解决输入输出长度不确定的问题。seq2seq属于encoder-decoder结构的一种，这里看看常见的encoder-decoder结构，基本思想就是利用两个RNN，一个RNN作为encoder，另一个RNN作为decoder。encoder负责将输入序列压缩成指定长度的向量，这个向量就可以看成是这个序列的语义，这个过程称为编码，如下图，获取语义向量最简单的方式就是直接将最后一个输入的隐状态作为语义向量c。也可以对最后一个隐含状态做一个变换得到语义向量，还可以将输入序列的所有隐含状态做一个变换得到语义变量。

## 缺陷

由于Encoder---c---Decoder中间的c向量长度固定，导致输入序列过长时会导致信息丢失，使用Attention机制解决这一问题。

## Attention

注意力机制是一种让模型重点关注关键信息并提取其特征用于学习分析的策略。最早将注意力机制引进NLP领域的是机器翻译等基于Encoder-Decoder框架的场景。传统的Encoder-Decoder框架有个问题：一些跟当前任务无关的信息都会被编码器强制编码进去，尤其输入很长或信息量很大时这个问题特别明显，进行选择性编码不是框架所能做到的。

对于语言模型，为了模型能够更加准确地判断，需要对输入的文本提取出关键且重要的信息。对输入文本的每个单词赋予不同的权重，携带关键重要信息的单词偏向性地赋予更高的权重。抽象来说，即是：对于输入Input，有相应的query向量和key-value向量集合，通过计算query和key关系的function，赋予每个value不同的权重，最终得到一个正确的向量输出Output。

注意力机制最早是为了解决Seq2Seq问题的，后来研究者尝试将其应用到情感分析、句对关系判别等其他任务场景，譬如关注aspect的情感分析模型ATAE LSTM、分析句对关系的ABCNN。

## Transformer

在Seq2Seq场景，注意力机制的引入，显著提升了模型的能力。但基于RNN的Seq2Seq框架有个很大的不足就是编码阶段必须按序列依次处理。于是，Google提出了Transformer，完全抛弃了CNN和RNN，只基于注意力机制捕捉输入和输出的全局关系，框架更容易并行计算，在诸如机器翻译和解析等任务训练时间减少，效果提升。Transformer模型最早是由Google于2017年在“Attention is all you need”一文中提出，在论文中该模型主要是被用于克服机器翻译任务中传统网络训练时间过长，难以较好实现并行计算的问题。后来，由于该方法在语序特征的提取效果由于传统的RNN、LSTM而被逐渐应用至各个领域，如GPT模型以及目前Google最新出品的新贵Bert模型等。相较于传统的RNN、LSTM或注意力集中机制，Transformer模型已经抛弃了以往的时序结构，更准确的来说其实际上是一种编码机制，该编码同时包括了语义信息（Multi-Head Attention）和位置信息（Positional Encoding）。

## SimpleNMT

本项目使用的翻译框架SimpleNMT是基于优秀的神经机器翻译系统（fairseq、OpenNMT、Tensor2Tensor等）所构建而成。将这些开源系统中实现复杂、冗余代码去除，保留核心代码和框架组合而成，非常易于初学者进行学习。

# 第三章

# 机器翻译模型实践和文献机器翻译工程

## 3.1机器翻译模型实践

### 3.1.0 数据集准备

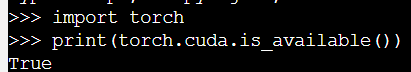
使用来自于https://github.com/openKG-field/PatentGLUE的专利机器翻译数据集。该数据集是来自专利领域的语料库制作的相关的双语平行数据集。该数据集在专利语言理解测评基准上进行打造，包括代表性的数据集、基准(预训练)模型、语料库、排行榜等，具有质量高，对应精准，数量庞大等优点，非常适合用来作为本项目等数据集。

### 3.1.1 模型训练

#### 3.1.1.1 配置环境

该模型基于Pytorch1.8.0、torchtext0.9.0，另外为加快训练速度，还需配置对应的CUDA版本。

搭建完成后，需检测环境是否配置成功。



#### 3.1.1.2 模型训练

|  |
| --- |
| $ python train.py -src zh -tgt en -data\_path content/dataset/ -save\_path content/model -batch\_size 64 |

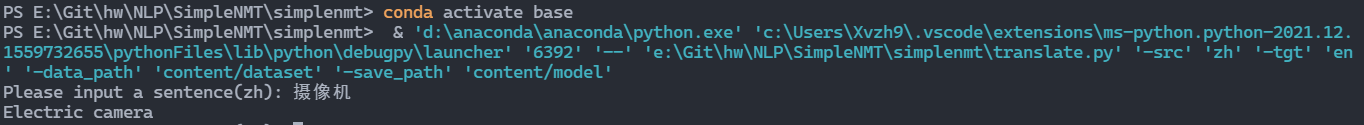
由于硬件条件的限制，batch\_size为64。学习率等超参数均使用默认值。

模型即可开始训练。



#### 3.1.1.3 模型的使用

在训练完成后可以使用模型进行翻译。可直接调用simpleNMT中的translate.py进行翻译。

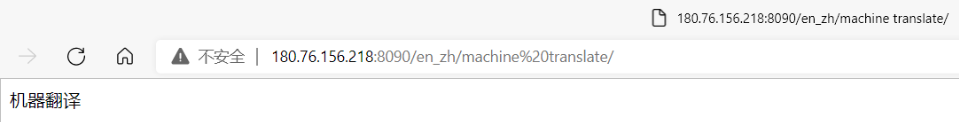


## 3.2 文献机器翻译工程

### 3.2.1建立翻译API

由于simpleNMT中的translate.py不可以直接被调用，于是在其基础进行修改，利用Flask创建类似于百度翻译API的接口，供后台调用。

最后效果图如下：



### 3.2.2 后台搭建

#### 3.2.2.1 后台架构

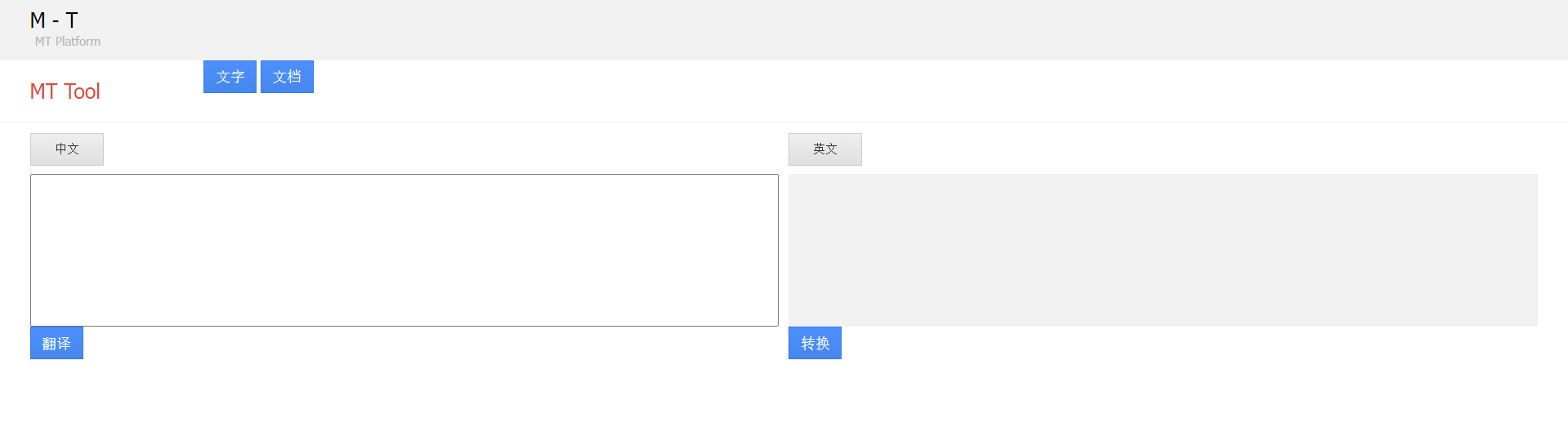
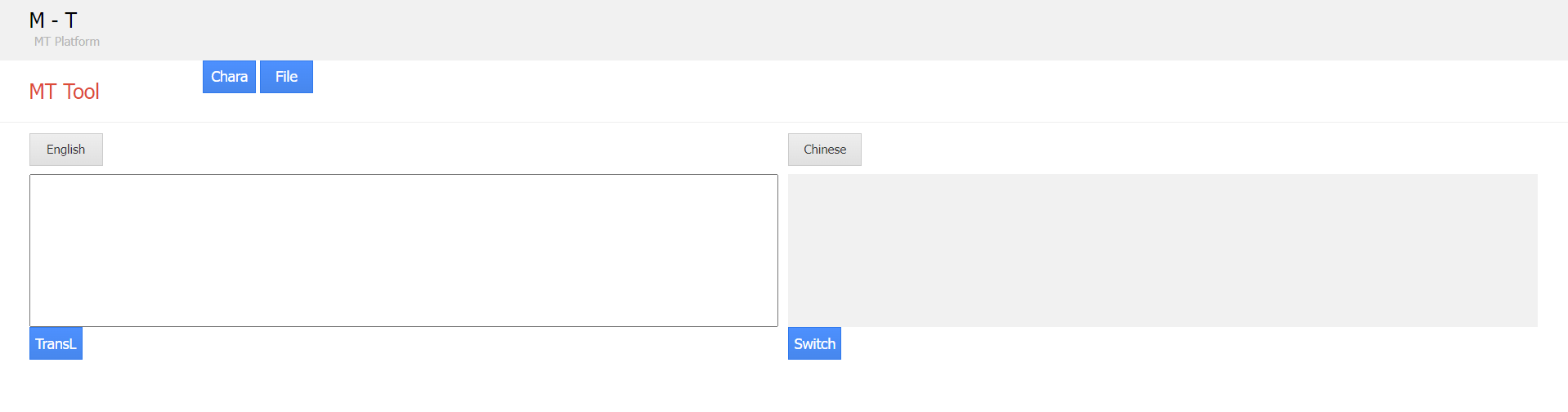
采用B/S架构进行后台搭建，调用训练模型API进行中英翻译并在前端界面显示。

训练模型API调用：





前端界面：



#### 3.2.2.2 Elasticsearch + Kibana架构

采用Elasticsearch和Kibana对中英数据集进行存储。

Elasticsearch引擎：

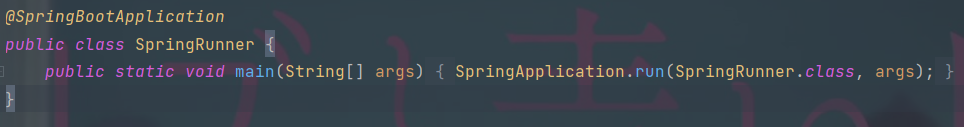
Kibana可视化界面 (已存储中英数据集)：



#### 3.2.2.3 Springboot + Elasticsearch 架构

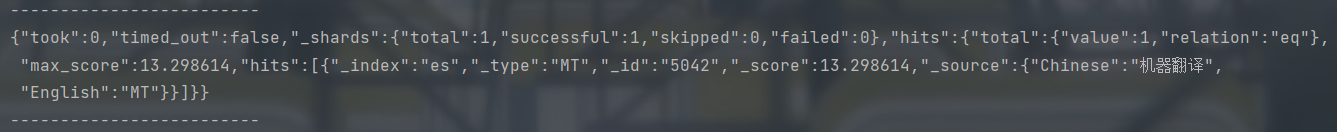
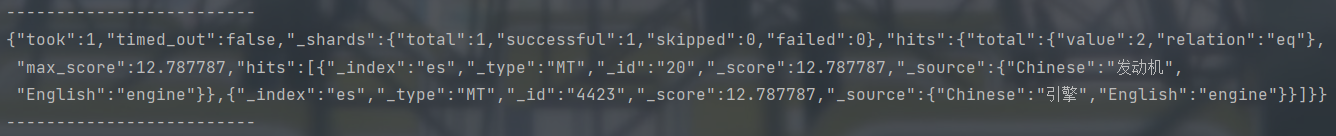
采用Springboot集成Elasticsearch并通过调用多元匹配方法对存储于 Elasticsearch中的数据集进行中英词汇匹配，并返回查询结果与训练模型API进行比对。

Springboot集成Elasticsearch：



调用Elasticsearch多元匹配方法：

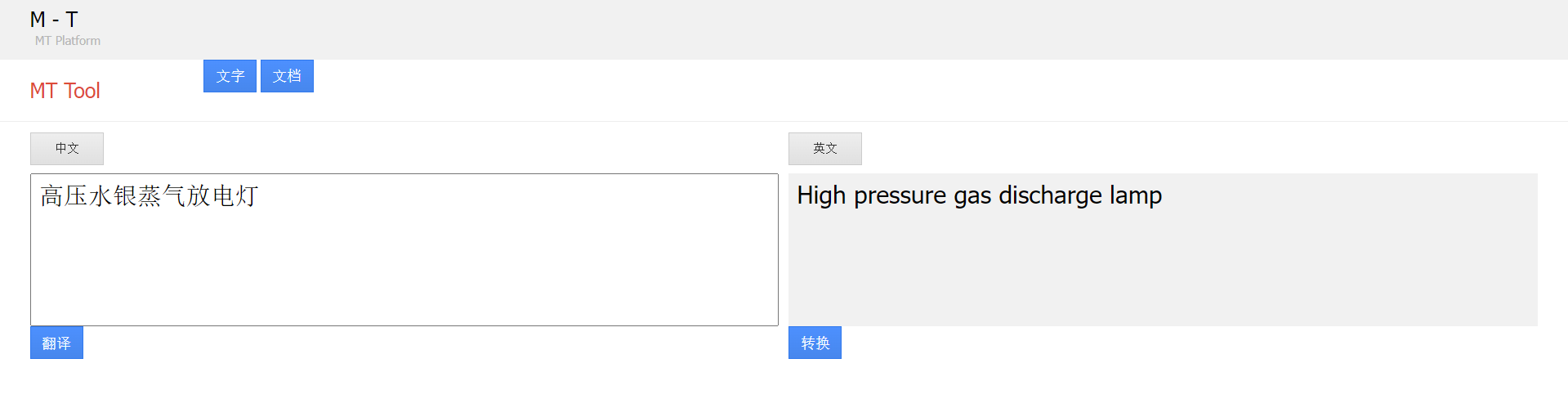
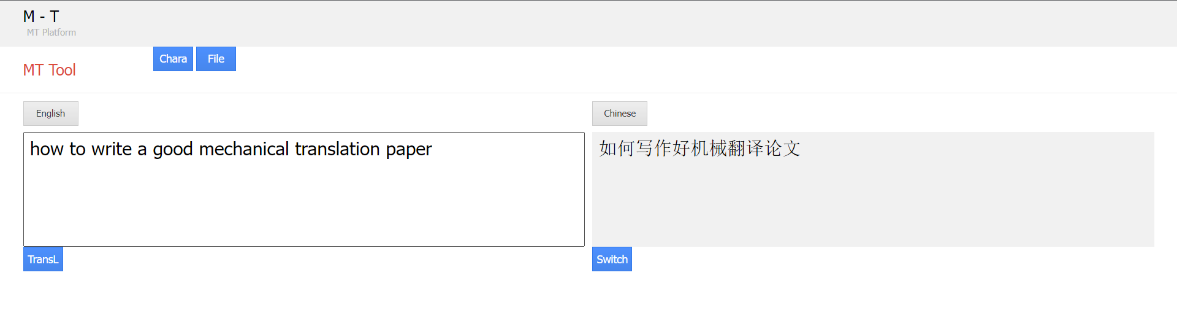
返回匹配结果，并与训练模型API进行比对：



### 3.2.3 项目成果展示

中文输入“高压水银蒸气放电灯”，返回结果是“High pressure gas discharge lamp”

英文输入“How to write a good machine translation paper”，返回结果是“如何写好机器翻译论文”



# 第四章

# 项目总结与讨论

## 4.1项目总结

通过本次项目实践，我们了解了NLP的一些基础知识，锻炼了我们的科研实践能力。在项目中，我们训练得到了一个中英文互译的模型和构建了一个基于Web前后端的实时翻译系统。但是模型的准确率有待提高，在之后的学习中可以通过搜集更多的训练数据和改善模型结构来提升准确率,并且采用Axure交互原型软件进一步的开发更多界面和功能。

## 4.2项目收获

本项目全程是以团队协作的方式进行，我们领悟到了团队成功运转的四大要素：1.选择好队友。2.为他们分配正确的工作；3.保持他们的积极性；4.帮助团队凝聚起来并保持团队的凝聚力。

《把信送给加西亚》里面说到，别人没有叫我们就主动去想事情做，那样获得的回报是令人敬佩和羡慕的，别人叫一次我们就去做，那样获得的回报是可观的，别人叫两次我们才去做，那样获得的回报是社会平均水平，别人叫三次才去做，获得的回报是微薄的，别人叫了多次才去做，那样不仅仅没有回报，反而会被人责骂。但是最终结果都要去做事情，我们何不主动些呢？有些人就等着分配任务，然后完成任务，分享成果，根本没有去为团队着想，为自己着想，这种人只能永远被人利用。有些人会想办法改进，并且经常有好的创意，这种人最终会获得成功，因为他主动了。我们选择的人要选择比较主动，能够及时领悟团队的任务并且及时去做甚至改善的人。所以选择队友是一门学问。

分配工作是很容易的，但是要做到正确分配工作就有学问了。我们进行分配任务的时候，刚开始是任由我们自己决定要做什么，然后再讨论进行整理，后来发现这样子任务并不能分配的较为合理。接着就是组长或者老师给组员配任务，后来发现，这样子分配的任务组员有些接受不了，例如时间，技术难度等等，久而久之往往会有畏难情绪。所以总结了以往的经验，分配我们采取了宏观控制加上微观调整。首先我们明确的知道每个组员的各项知识技能水平。组员对感兴趣的模块会说的比较多，比较有见解。接着再根据他们的以往的编程经验，快速定出两周内要完成的任务，不能定得太过遥远，因为项目刚刚开始时无法预知进度的。

人的积极性是个很有趣的东西，把握拿捏不好分寸，就会影响到一个人做事情的积极性。引用一段很经典的X,Y理论：X,Y理论可以用来分析一个经理的管理行为，如果这个经理的思想是X理论的思想，那么他就会采取比较强硬的措施，例如军队式的管理方法。这种经理就会用各种严格的规章制度来强迫人们进行工作，触犯了规章制度就会受到经济上的惩罚。我认为这种管理方式只适合一些标准化操作程度比较高的企业，对于软件行业来说，人才得到了极大的重视，近期人才流动性较大，员工的个性也比较强烈，无法应用这种强硬的措施。人较低需求层次是生存，高层次是自我实现，在软件行业中，生存已经不是问题，每个人都抱着自我实现的目标而奋斗，所以我认为在软件行业中采用Y理论比较适合，即一定要让员工参与到决策中。所以要让大家有积极性，就要让人感觉，这个项目是大家努力的结果，是大家团结的结果，而不是某一个技术牛人的结果。

光光人的积极性还不够，每个人的用力方向还必须一样才能发挥团队的能量，团队是有一个生命周期的，分别是形成，躁动，稳定，表现。当团队形成的时候，一切都是新鲜的。因为团队形成的时候，大家的工作热情比较高涨，但是容易出现焦虑，因为在团队中，他们的定位都不明确，所以作为团队的管理者在这个阶段应该做的是建立团队文化，例如团队名称等等，甚至连什么时候开会，什么时候往也要事先有个规划，然后让团队的人之间互相充分的了解，然后做一些合作的工作来让大家了解在这个团队里面应该如何进行沟通协作。然后团队渐渐稳定，生产力不断提高，项目和队员的能力稳步增长，目标清晰，成员之间已经形成一个整体，那么到这个阶段项目经理就比较轻松了，只要进行一些团队之间的关系处理，以及让大家进行一些重大决策就可以了。最后一个阶段就是表现阶段，那么这个阶段我认为是丰收阶段，团队成员配合已经到达非常默契的程度，基本上不需要频繁的交流，一切都有现成的流程，效率空前的提高。

虽然最后我们的项目成功实现了，但是在这几个月来也是走了不少弯路，在项目的进展过程中暴露出很多的问题，我们还需要不断提升我们的工程素养和项目代码能力，非常感谢长期支持我们的老师和同学。

# 参考文献：

**[1]** BrownPF，CookeJ．DellaSA，eta1．A StatisticalApproach to Machine Translation．Computational Linguistics，1990．16(2)：79 ～ 85

**[2]** Nagao M.A Framework of a mechanical translation between Japanese and English by analogy principle．Artificial and Human Intelligence，1984．173～ 180

[3] 戴新宇, 尹存燕, 陈家骏, 等. 机器翻译研究现状与展望[J]. 计算机科学, 2004, 31(11):177~184.

[4] 肖桐, 朱靖波. 机器翻译：基础与模型[M]. 2020. 东北大学自然语言处理实验室·小牛翻译 :1～748

[5] CSDN打工小黄人. 深度学习NLP的各类模型及应用总结[EB/OL]. [2019-03-31 ]. <https://blog.csdn.net/huanghaocs/article/details/88932072.>

[6]  打不死的小强easyai-tech. Encoder-Decoder 和 Seq2Seq[EB/OL]. [2019-10-28]. <https://easyai.tech/ai-definition/encoder-decoder-seq2seq/.>

[7] 简书·骆旺达. Seq2Seq模型概述[EB/OL]. 2019.01.12[2021-12-27]. Seq2Seq模型概述.https://www.jianshu.com/p/b2b95f945a98.