

基于对偶分解的词语对齐搜索算法

沈世奇, 刘洋, 孙茂松

(智能技术与系统国家重点实验室; 清华信息科学与技术国家实验室;
清华大学 计算机系, 北京 100084)

摘要: 词语对齐旨在计算平行文本中词语之间的对应关系, 对机器翻译、双语词典构造等多项自然语言处理任务都具有重要的影响。虽然近年来词语对齐在建模和训练算法方面取得了显著的进展, 但搜索算法往往都采用简单的贪心策略, 面临着搜索错误较大的问题。本文提出了一种基于对偶分解的词语对齐搜索算法, 将复杂问题分解为两个相对简单的子问题, 迭代求解直至收敛于最优解。由于对偶分解能够保证求解的收敛性和最优性, 本文提出的搜索算法在 2005 年度 863 计划词语对齐评测数据集上显著超过 GIZA++ 和判别式词语对齐系统, 对齐错误率分别降低 4.2% 和 1.1%。

关键词: 词语对齐; 判别式模型; 搜索算法; 对偶分解

Search for Discriminative Word Alignment via Dual Decomposition

Shiqi Shen, Yang Liu and Maosong Sun

(Department of Computer Science and Technology, State Key Lab on Intelligent Technology and Systems, Tsinghua University, Beijing 100084, China)

Abstract: Word alignment aims to calculate the corresponding relationship between the words in parallel texts. It has important influence on machine translation, bilingual dictionary construction and many other natural language processing tasks. Although in recent years the word alignment has made significant progress in modeling and training algorithm, its search algorithm often uses simply greedy strategies and faces the problem of large search errors. This paper proposed a word alignment search algorithm based on dual decomposition, making a complex problem into two relatively simple sub-problems and iteratively solving it until convergence to the optimal solution. For the dual decomposition can ensure the convergence and optimality of solutions, this algorithm significantly exceeds GUZA++ and discriminant word alignment system on alignment error rates when testing on the 863 Projects word alignment evaluation data set of 2005. Alignment error rate is decreased by 4.2% and 1.1% respectively.

Key words: word alignment; discriminative model; Search Algorithm; dual decomposition

1 引言

词语对齐旨在计算平行文本中词语之间的对应关系。例如, 给定一个中文句子和一个英文句子¹

中国建筑业对外开放呈现新格局。

The Opening of China ' s construction industry to the outside presents a new structure.

收稿日期: **定稿日期:**

基金项目: 863 计划项目 (批准号: 2012AA011102 和 2011AA01A207); 媒体与网络技术教育部-微软重点实验室项目 (编号 20123000007)

作者简介: 沈世奇 (1990—), 男, 博士生, 主要研究方向为统计机器翻译; 刘洋 (1979—), 男, 副研究员, 主要研究方向为机器翻译、自然语言处理; 孙茂松 (1962—), 男, 教授, 主要研究方向为自然语言处理。

¹ 该例句及词语对齐源自(Liu et al., 2010)

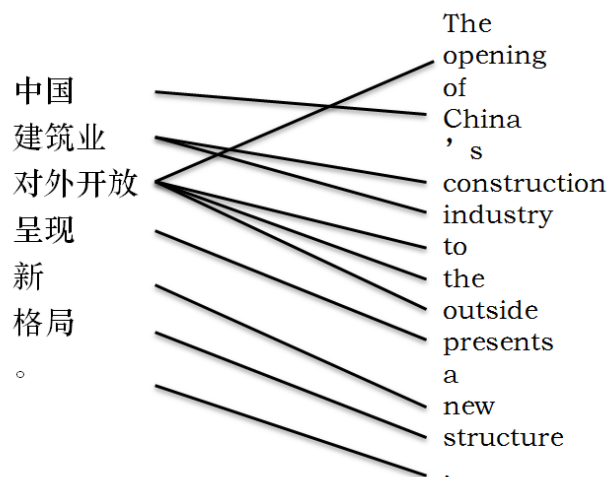


图 1: 词语对齐示例

图 1 给出了这两个句子的词语对齐。由于中文词“中国”与英文词“China”互相翻译，两者之间存在一条连线显示对应关系。需要指出的是，中文词与英文词并不完全是一一对应，如中文词“建筑业”对应两个英文词“*construction industry*”。更复杂的情况是一个中文词对应到若干个非连续的英文词，如“对外开放”对应英文词“*opening ... to the outside*”。因此，由于自然语言的多样性和复杂性，发现不同语言之间的词语对应关系是一个非常具有挑战性的问题。

词语对齐最早是作为机器翻译的中间结果由[2]Brown et al. (1993)提出。由于词语对齐能够为不同自然语言在词语层建立关联，它在机器翻译、机器辅助翻译、词义消歧和双语词典构建等多项自然语言处理任务中起着关键的作用。例如，在统计机器翻译中，无论是基于短语的方法[8] (Koehn et al., 2003)、基于层次短语的方法[4] (Chiang, 2007)还是基于句法的方法[7] (Galley et al., 2006)，往往都依赖于词语对齐进行规则抽取，因而词语对齐的质量对机器翻译译文的质量具有重要的影响。

词语对齐模型大致可分为生成式模型和判别式模型两大类。生成式模型([2]Brown et al., 1993; [22]Vogel et al., 1996; [11]Liang et al., 2006)为词语对齐设计生成故事(generative story)，其优点在于不需要标注数据进行训练，很容易应用于各个领域，其缺点在于模型难以扩展。判别式模型([12]Liu et al., 2005; [15]Moore, 2005; [1]Blunsom and Cohn, 2006; [13]Liu et al., 2010)将各种知识源作为特征函数加入到模型中，易于扩展，其主要缺点在于依赖于标注数据。在训练算法方面，生成式模型主要使用 EM 算法[2] (Brown et al., 1993)实现无监督学习，而判别式模型可以使用最小错误率训练算法[17] (Och, 2003)实现直接优化评价指标，大幅提高对齐质量。[6]Dyer et al. (2011) 最近提出了判别式模型的无监督训练，实现了判别式模型与无监督学习的优势互补。

然而，尽管词语对齐在建模和训练算法方面取得了较大的进展，多数搜索算法仍然面临着搜索错误严重的问题。给定一个源语言句子 f_1' 和一个目标语言句子 e_1' ，可能的词语对齐结果数量是 $2^{J \times J'}$ 。对于以 IBM 模型[2] (Brown et al., 1993)为代表的生成模型而言，虽然简单的模型 1 和模型 2 可以精准计算 Viterbi 对齐，但是对于更复杂的模型 3、4、5 只能使用爬山法计算近似解([2]Brown et al., 1993; [16]Och and Ney, 2003)。对于以线性模型[13] (Liu et al., 2010)为代表的判别式模型而言，往往也只能使用柱搜索算法来计算近似解。

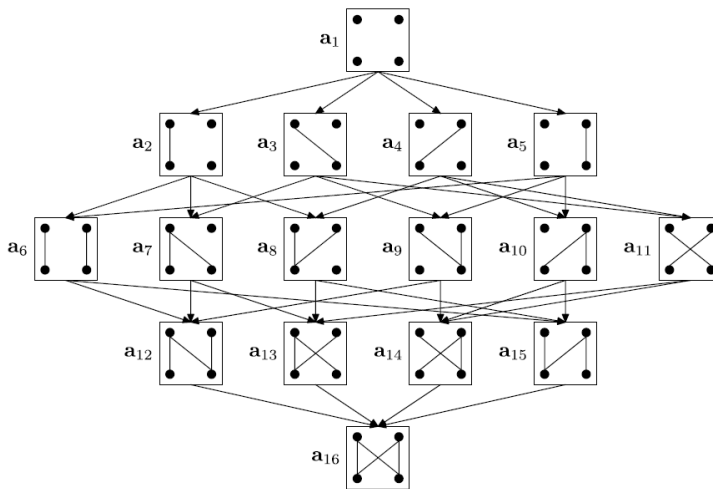


图 2: 词语对齐的搜索空间, 此图源自[13] (Liu et al., 2010)。对于长度分别为 J 和 I 的平行句对, 可能的词语对齐数量是 $2^{J \times I}$

图 2 给出了[13]Liu et al. (2010)所提出的柱搜索算法的搜索空间, 该算法以空对齐为起点, 不断添加连线直至模型分数无法增加为止。由于词语对齐中连线之间存在错综复杂的依赖关系, 无论是爬山法还是柱搜索算法都面临着严重的搜索错误问题。最近, [19]Riesa and Marcu (2010)将立方体剪枝[4] (Chiang, 2007)引入词语对齐, 但本质上仍然只能计算近似解。因此, 搜索算法已经成为制约词语对齐质量的瓶颈问题。

近年来, 对偶分解(dual decomposition)被广泛应用于句法分析[10] (Koo et al., 2010)和机器翻译[3] (Chang and Collins, 2011)等自然语言处理任务来实现精准求解(exact decoding), 均取得了良好的效果。为了缓解词语对齐所面临的搜索错误问题, 本文提出一种基于对偶分解的词语对齐搜索算法, 其基本思想是将复杂的问题分解为两个相对简单的子问题, 迭代求解直至收敛。由于对偶分解能够确保求解的收敛性和最优性[20] (Rush and Collins, 2011), 我们的方法在 2005 年度 863 计划词语对齐评测数据集上显著超过 IBM 模型的爬山法[2] (Brown et al., 1993)与判别式模型的柱搜索算法[13] (Liu et al., 2010), 对齐错误率分别降低 4.4% 和 1.1%。

2 基于对偶分解的词语对齐搜索算法

对偶分解的基本思想是将一个复杂的问题分解为两个相对简单的子问题, 之后对两个子问题进行分别求解, 通过一个数据结构记录两者的差异之处并利用该数据结构试图使两个子问题的解趋于一致。当算法经过多轮迭代收敛 (即两个子问题得出相同的解) 时, 该解为最优解。

图 3 给出了基于对偶分解的词语对齐搜索算法的一个示例。给定中文句子“呈现 新格局。”和英文句子“present a new structure.”, 假设我们将一个词语对齐模型拆成两个子模型进行分别求解, 并定义函数 $u(j, i)$ 来记录两个子模型的解的差异之处。初始状态对于任意的 i 和 j , $u(j, i)$ 均设为 0。图 3(a)显示了两个子问题的解, 上面是子问题一的解, 下面是子问题二的解。可以发现, 两个解存在差异: 子问题一的连线 $(2, 2)$ 与子问题二的连线 $(2, 1)$ 不一致, 即子问题一将“新”与“a”连线而子问题二将“呈现”与“a”相连。因此, 算法更新 u 函数, 将 $u(2, 2)$ 设为 -1, 将 $u(2, 1)$ 设为 1, 试图使两个子问题的解趋向一致 (如图 3(b)所示)。再次进行求解, 两个子问题的解达成一致, 获得最终解 (如图 3(c)所

示)。这也说明了对偶分解的方法得到的解 3(c)能够优于子问题的解 3(a)的原因。

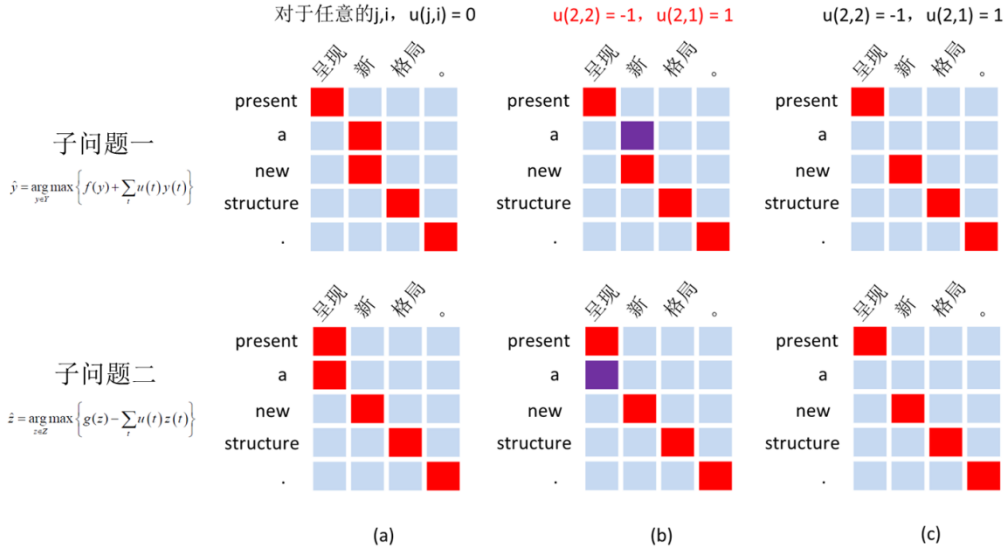


图 3: 基于对偶分解的词语对齐算法示例。对偶分解将一个复杂问题分解为两个子问题, 分别求解并使用惩罚函数 $u(j, i)$ 记录两个解的差异之处, 使得两个子问题的解趋近一致, 最终收敛于最优解。

更形式化地, 给定源语言句子 f_1' 和目标语言句子 e_1' , 基于对偶分解的词语对齐的搜索算法的目标是

$$\arg \max_{y \in Y, z \in Z} (f(y) + g(z)) \quad (1)$$

同时满足对任意 $(j, i) \in \{1, \dots, J\} \times \{1, \dots, I\}$

$$y(j, i) = z(j, i) \quad (2)$$

其中, Y 和 Z 分别是两个词语对齐的解集合, y 和 z 分别是其中的一个解, $f(y)$ 对应子问题一的模型, $g(z)$ 对应子问题二的模型, $y(j, i) = 1$ 表示子问题一将 f_j 与 e_i 连在一起, 0 则表示没有相连。因此, 对偶分解的目标是计算两个子问题达成一致的最优解。

对偶分解将按照子问题一和子问题二进行分别求解。子问题一的优化目标是

$$\arg \max_{y \in Y} \left(f(y) + \sum_{j, i} u(j, i) y(j, i) \right) \quad (3)$$

子问题二的优化目标是

$$\arg \max_{z \in Z} \left(g(z) - \sum_{j, i} u(j, i) z(j, i) \right) \quad (4)$$

图 4 给出了基于对偶分解的词语对齐搜索算法。算法的输入是源语言句子 f_1' 和目标语言句子 e_1' (第 1 行)。首先初始化惩罚函数 $u(j, i)$, 全部设为 0 (第 2 行)。之后进入迭代求解 (第 3 行), 分别对子问题一 (第 4 行) 和子问题二 (第 5 行) 进行求解。如果满足收敛条件, 即两个子问题生成完全相同的对齐, 则返回最优解 (第 6 至 7 行), 否则更新惩罚函数 (第 9 行) 进入下一轮迭代。如若超过最大迭代次数仍然没有收敛, 则选取迭代过程中子问题的最优解作为最后的结果。其中, α_k 是第 k 轮迭代的更新比率, 用来控制算法收敛

的速度。[21]Rush and Collins (2012)明当 α_k 满足以下条件时

```

1: procedure ALIGN( $f_1^J, e_1^I$ )
2:    $u^{(1)}(j, i) \leftarrow 0, (j, i) \in \{1, \dots, J\} \times \{1, \dots, I\}$ 
3:   for  $k = 1, \dots, K$  do
4:      $y^{(k)} \leftarrow \arg \max_{y \in \mathcal{Y}} \left( f(y) + \sum_{j,i} u^k(j, i) y(j, i) \right)$ 
5:      $z^{(k)} \leftarrow \arg \max_{z \in \mathcal{Z}} \left( g(z) - \sum_{j,i} u^k(j, i) z(j, i) \right)$ 
6:     if  $y^k(j, i) = z^k(j, i)$  for all  $(j, i)$  then
7:       return  $(y^{(k)}, z^{(k)})$ 
8:     else
9:        $u^{(k+1)}(j, i) \leftarrow u^{(k)}(j, i) - \alpha_k (y^{(k)}(j, i) - z^{(k)}(j, i))$ 
10:    end if
11:  end for
12: end procedure

```

▷ 初始化惩罚函数
 ▷ 迭代求解
 ▷ 求解子问题一
 ▷ 求解子问题二
 ▷ 算法收敛
 ▷ 得到最优解
 ▷ 更新惩罚函数

图 4: 基于对偶分解的词语对齐搜索算法。

$$\lim_{k \rightarrow \infty} \alpha_k = 0 \quad (5)$$

并且

$$\sum_{k=1}^{\infty} \alpha_k = \infty \quad (6)$$

对偶分解算法能确保评价函数收敛。

3 实验

本文的训练集是中英 24 万句对的政府新闻类双语语料库, 开发集和测试集是 2005 年度 863 计划词语对齐评测数据集, 开发集包含 502 句对, 测试集包含 505 句对。我们所使用的评价指标是由[16]Och and Ney (2003)的对齐错误率(Alignment Error Rate, 简称 AER), 对齐错误率越低表示对齐质量越高。

我们与以下两个系统进行对比:

1. GIZA++: IBM 模型[2] (Brown et al., 1993), 采用 EM 算法进行参数训练, 爬山法进行搜索, 由[16]Och and Ney(2003)实现。
2. Vigne: [13]Liu et al. (2010)提出的判别式词语对齐方法, 采用最小错误率训练算法 [17] (Och, 2003)进行参数训练, 柱搜索算法进行搜索。

我们首先使用 GIZA++的默认设置进行双向训练, 得到中英和英中两个方向的 IBM 模型 4 的对齐结果以及模型参数。Vigne 主要采用 GIZA++输出的 IBM 模型参数作为主要特征, 具体为[13] (Liu et al. 2010)提出的 9 个特征, 利用最小错误率训练(Och, 2003)在开发集上优化特征权重, 在测试集上采用柱搜索算法获得最优对齐结果。我们的词语对齐系统在建模和训练算法上与 Vigne 完全一致, 只是在搜索算法上采用对偶分解法。

系统	模型	训练	搜索	设置	AER
GIZA++	IBM 模型 4	EM	爬山法	中英	24.4
				英中	25.3
				交集	23.9
				并集	29.1
Vigne	线性判别式模型	最小错误率训练	柱搜索		20.8
我们的工作	线性判别式模型	最小错误率训练	对偶分解		19.7

表 1: GIZA++、Vigne 与我们的方法的对比结果。对齐错误率 AER 越低表示对齐质量越高。我们采用了[9]Koehn (2004)提出的显著性检验方法验证我们的系统与 GIZA++和 Vigne 的差别在统计上是显著的($p=0.05$)。

α_k	$\frac{1}{t+1}$	$\frac{1}{k}$	0.1	0.01	0.001
收敛率	0.9644	0.9663	0.6040	0.9069	0.7525
总体 AER	0.1995	0.1993	0.2048	0.2022	0.2036

表 2: 迭代更新比率 α_k 对结果的影响

3.1 与传统搜索算法的对比

表 1 给出了我们的系统与 GIZA++和 Vigne 的对比结果。GIZA++产生四种对齐结果:

1. 中英: 以中文为源语言, 英文为目标语言, 使用 IBM 模型 4;
2. 英中: 以英文为源语言, 中文为目标语言, 使用 IBM 模型 4;
3. 交集: 两个方向对齐结果的交集;
4. 并集: 两个方向对齐结果的并集。

其中, GIZA++的交集取得了四种设置中最好的结果 23.9。判别式词语对齐系统 Vigne 取得的对齐错误率是 20.8, 显著超过了 GIZA++。而我们的对偶分解算法取得了最好的对齐错误率 19.7, 超过 GIZA++4.4 个百分点, Vigne1.1 个百分点。我们采用了[9]Koehn(2004)提出的显著性检验方法验证上述差别在统计上是显著的($p=0.05$), 从而验证了对偶分解算法确实有效减少了搜索错误。

在搜索效率上, GIZA++作为一个无监督的方法, 需要对较大的数据进行整体的处理, 并不能对单句进行词对齐, 与 Vigne 和我们的方法很难比较。

而我们使用的对偶分解采用的是迭代的方法, 每一轮迭代都需要对两个子问题分别搜索, 所以在时间上要慢于 Vigne, 时间复杂度随着迭代次数的增长而增长。

3.2 迭代更新比率 α_k 对收敛率和对齐错误率的影响

[21]Rush and Collins (2012)指出迭代更新比率 α_k 将对对偶分解的迭代是否收敛产生重要影响。因此，我们通过实验研究迭代更新比率对收敛率的影响。所谓收敛率，是指待对齐文本中达到收敛的句子比例。例如，如果对偶分解算法在100个句子中的90个句子达到收敛，则收敛率为90%。我们设定最大迭代次数K为100，当迭代超过100次后我们认为该句子在本方法下不收敛。表2给出了不同的迭代更新比率及相应的收敛率和对齐错误率。其中，k为迭代次数，t是一个整数，其值为在迭代过程中对偶分解的评价函数 $L(u) = \max_{y \in Y, \epsilon \in Z} L(u, y, \epsilon)$ 增长的次数。由于 $1/(t+1)$ 和 $1/k$ 满足公式(5)和(6)，它们对应的收敛率均超过96%并取得较低的AER。与之相反，常数型的更新比率(如0.1、0.01和0.001)则对应较低的收敛率和较高的AER。因此，更新比率对于收敛率和AER具有重要的影响²。

3.3 句子长度对收敛率的影响

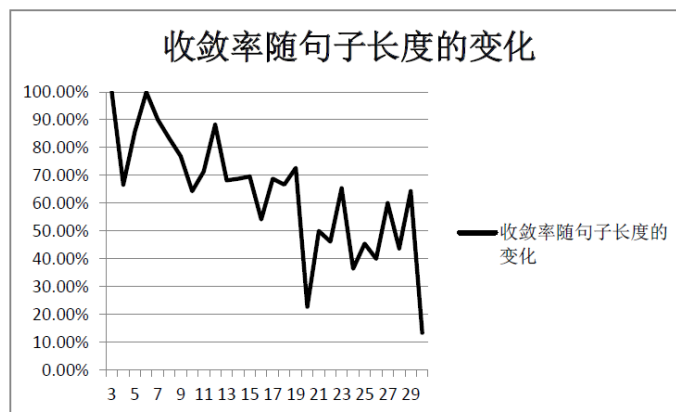


图5：句子长度对收敛率的影响。横坐标是句子长度，纵坐标是收敛率。

对于词语对齐问题来说，句子的长度直接影响了算法的复杂性。直观来看，句子越短，词语对齐所需要的搜索空间越小。反之，句子越长，搜索空间越大。我们设置不同的迭代收敛比率 α_k ，观察对偶分解算法在不同长度的句子上的收敛率。图5给出了在 $\alpha_k = 0.1$ 、总收敛率为0.6040的条件下，句子的收敛比率随句子长度变化的关系图。可以看出，对偶分解算法的收敛率随着句子长度的增长大致是呈现下降趋势。也就是说，由于句子长度的增长而增大的搜索空间会使得算法收敛率下降。

3.4 收敛与不收敛数据集上对偶分解与柱搜索的比较

表3给出了收敛数据集和不收敛数据集上对偶分解算法与柱搜索算法的比较情况。我们发现无论是在收敛数据集上还是在非收敛数据集上，对偶分解算法都取得了比柱搜索算法更好的结果，进一步证明对偶分解算法能有效地提高对齐质量³。

²需要指出的是，在实际过程中，我们发现对于较长的句子无法实现100%的收敛率，原因在于收敛性与最优性的约束条件并不完全一致。但即使对偶分解算法无法通过收敛获得最优解，也可以计算出较好的次优解。

³值得注意的是，在收敛数据集和非收敛数据集上对齐错误率有较大的差异，收敛数据集上的对齐错误率明显较优，原因在于对偶分解能够在相对简单的句子上收敛，故根据收敛性就会将数据集划分为“较难”和“较易”的两部分。

α_k	$\frac{1}{t+1}$	$\frac{1}{k}$	0.1	0.01	0.001
对偶分解在收敛数据集上的 AER	0.1977	0.1952	0.1848	0.1927	0.1856
柱搜索在收敛数据集上的 AER	0.2067	0.2038	0.1886	0.1982	0.1897
对偶分解在不收敛数据集上的 AER	0.2403	0.3134	0.2303	0.2818	0.2477
柱搜索在不收敛数据集上的 AER	0.2445	0.3340	0.2335	0.2932	0.2542

表 3: 收敛和不收敛数据集上对偶分解与柱搜索的比较

3.5 对偶分解与子模型比较

方法	参数	AER
子模型一	9 个特征	0.208
子模型二	6 个特征	0.202
对偶分解	子模型一与子模型二	0.197

表 4: 对偶分解与子模型的对齐错误率比较

在实验中,我们采用的子模型一是线性模型,采用[13] (Liu et al. 2010)提出的 9 个特征,与 Vigne 的特征一致,子模型二也是线性模型,采用[13] (Liu et al. 2010)提出的 6 个特征。表 4 给出了对偶分解算法与其使用的两个子模型各自的对齐错误率。我们观察到与两个子模型相比,对偶分解方法取得了更优的结果,这也进一步说明对偶分解算法的确能够在子模型的基础上减少搜索错误,降低对齐错误率。

4 相关工作

[10]Koo et al. (2010)首次将对偶分解应用于自然语言处理领域,他们在句法分析上验证了对偶分解算法的有效性。[20]Rush and Collins (2011)与[3]Chang and Collins (2011)则分别将对偶分解算法引入基于句法和基于短语的机器翻译中,有效减少了搜索错误。[5]DeNero and Macherey (2011)首次将对偶分解引入词语对齐,但他们的重点是模型融合,我们则侧重扩展判别式模型的搜索算法。[14]Martins et al. (2011)进一步将对偶分解扩展为可以处理若干个子问题。

除了对偶分解,整数线性规划(Integer Linear Programming)是另一项被广泛应用的优化技术。[18]Ravi and Knight (2010)将整数线性规划首次应用于词语对齐,为 GIZA++ 设计了精准搜索算法。

综合利用多个结果来改善结果的方法有模型融合和系统融合,但是对偶分解本质上是一种最优化搜索算法,它可以被用于系统融合:不是将原问题分成两个子问题,而是分成多个子问题。

5 结论

本文提出了一种基于对偶分解的词语对齐搜索算法,通过子问题迭代分别求解直至收敛于最优解。我们的方法在 2005 年度 863 计划词语对齐评测数据集上显著超过生成式模型的爬山法和判别式模型的柱搜索算法。未来我们将深入研究提高长句收敛率的方法,希望能进一步减少词语对齐的搜索错误。

参考文献

- [1] Phil Blunsom and Trevor Cohn. 2006. Discriminative word alignment with conditional random fields. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, ACL-44, pages 65–72, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [2] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311, June.
- [3] Yin-Wen Chang and Michael Collins. 2011. Exact decoding of phrase-based translation models through lagrangian relaxation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, pages 26–37, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [4] David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.
- [5] John DeNero and Klaus Macherey. 2011. Model-based aligner combination using dual decomposition. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 420–429, Portland, Oregon, USA, June. Association for Computational Linguistics.
- [6] Chris Dyer, Jonathan H. Clark, Alon Lavie, and Noah A. Smith. 2011. Unsupervised word alignment with arbitrary features. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 409–419, Portland, Oregon, USA, June. Association for Computational Linguistics.
- [7] Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In Proceedings of COLING-ACL 2006, pages 961–968, Sydney, Australia, July.
- [8] Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In Proceedings of HLT-NAACL 2003, pages 127–133, Edmonton, Canada, May.
- [9] Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, Proceedings of EMNLP 2004, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.
- [10] Terry Koo, Alexander M. Rush, Michael Collins, Tommi Jaakkola, and David Sontag. 2010. Dual decomposition for parsing with non-projective head automata. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10, pages 1288–1298, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [11] Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, HLT-NAACL '06, pages 104–111, Stroudsburg,

- PA, USA. Association for Computational Linguistics.
- [12] Yang Liu, Qun Liu, and Shouxun Lin. 2005. Log-linear models for word alignment. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05, pages 459–466, Stroudsburg, PA, USA. Association for Computational Linguistics.
 - [13] Yang Liu, Qun Liu, and Shouxun Lin. 2010. Discriminative word alignment by linear modeling. *Comput. Linguist.*, 36(3):303–339.
 - [14] Andre Martins, Noah Smith, Mario Figueiredo, and Pedro Aguiar. 2011. Dual decomposition with many overlapping components. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 238–249, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
 - [15] Robert C. Moore. 2005. A discriminative framework for bilingual word alignment. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05, pages 81–88, Stroudsburg, PA, USA. Association for Computational Linguistics.
 - [16] Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51, March.
 - [17] Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.
 - [18] Sujith Ravi and Kevin Knight. 2010. Does giza++ make search errors? *Computational Linguistics*, 36(3).
 - [19] Jason Riesa and Daniel Marcu. 2010. Hierarchical search for word alignment. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10, pages 157–166, Stroudsburg, PA, USA. Association for Computational Linguistics.
 - [20] Alexander M. Rush and Michael Collins. 2011. Exact decoding of syntactic translation models through lagrangian relaxation. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, pages 72–82, Stroudsburg, PA, USA. Association for Computational Linguistics.
 - [21] Alexander Rush and Michael Collins. 2012. A tutorial on lagrangian relaxation and dual decomposition for nlp. *Journal of Artificial Intelligence Research*.
 - [22] Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In Proceedings of the 16th conference on Computational linguistics - Volume 2, COLING '96, pages 836–841, Stroudsburg, PA, USA. Association for Computational Linguistics.

作者联系方式：姓名 地址 邮编 电话（最好手机） 电子邮箱

沈世奇 北京清华大学 FIT 楼 4-506 100084 15120001227 vicapple22@gmail.com