机器翻译评测中的模糊匹配*

刘洋· 刘群 林守勋·
中国科学院计算技术研究所数字化技术研究室
北京 100080

摘要：
目前，大多数机器翻译自动评测方法都没有考虑在未匹配的词语中可能包含被忽略的信息。本文提出一种在参考译文和待评测译文之间自动搜索模糊匹配词对的方法，并给出相似度的计算方法。模糊匹配和计算相似度的整个过程将通过一个例子进行说明。实验表明，我们的方法能够较好地找到被忽略的、有意义的词对。更重要的是，通过引入模糊匹配，BLEU的性能得到显著的提高。模糊匹配可以用来提高其他机器翻译自动评测方法的性能。
关键词：机器翻译评测；模糊匹配
中图分类号：TP391

Fuzzy Matching in Machine Translation Evaluation

LIU Yang LIU Qun LIN Shouxun
Digital Technology Lab, Institute of Computing
Technology ,Chinese Academy of Sciences
Beijing 100080

Abstract:
Most current automatic metrics of machine translation evaluation do not consider that among unmatched words there may be neglected information. In this paper, we describe a strategy to find fuzzy-matched word pairs between reference and candidate translations automatically and propose an approach to compute the similarity. The whole process of finding fuzzy-matched word pairs and computing their similarity is demonstrated in detail. Experiments show that our method is capable of finding neglected meaningful word pairs fairly well. More importantly, the performance of BLEU is significantly improved by integrating fuzzy matching. Fuzzy matching is possible to be utilized to improve other automatic methods.
Key words: machine translation evaluation; fuzzy matching

1. 前言

由于对机器翻译系统进行人工评测既代价高昂又耗时持久，机器翻译自动评测成为近年

* 本文的研究是在国家 863 项目“中文平台总体技术研究与基础数据库建设”（2001AA114010）的支持下完成的。
* 作者简介： 刘洋：男，1979 年生，博士研究生，研究方向是统计机器翻译、机器翻译自动评测。
刘群：男，1966 年生，副研究员，研究方向是自然语言处理、机器翻译、信息提取。
林守勋：男，1948 年生，研究员，研究方向是多媒体技术、分布协同计算。

来的一个研究热点。机器翻译自动评测方法的最重要的目标是对机器翻译的译文质量给出一个分值，而这个分值与人工评测的结果应当有极高的相关性[1]。

北京大学计算语言学研究所的俞士汶教授于七五期间主持开发MTE系统，该系统被认为是世界上第一个机器翻译自动评测系统。MTE以句子为单位，根据测试集进行评测。为了实现自动测试的目标，MTE还借鉴了语言测试中分离式测试的方法，即对每一个句子，不是评测整句的翻译，而是每句侧重一个测试点（每个测试点代表一个语言点），只评测测试点的翻译[2]。用测试集进行自动评测是一种有意义的尝试，这种方法使机器翻译评测摆脱了评测过程中的主观性，同时也节省了人力和物力。但是，自动评测系统测试集的建立是一项繁琐而且复杂的任务，需要机器翻译专家、机器翻译系统开发者、语言学家和软件工程师的密切合作。同时，建立测试集是一个长期的过程，测试点的确立与描述需要不断完善[3]。

目前，机器翻译自动评测的主流方法是基于N-gram共现的方法，其基本思想是计算系统输出译文（机器翻译系统生成）与若干个参考译文（人工翻译）之间的相似程度。BLEU和NIST是当前最常用的两种自动方法[4]。

然而，BLEU和NIST（也许还包括其他自动方法）的一个很严重的问题是只允许完全匹配。所谓完全匹配是指两个词或者完全相同或者完全不相同。基于完全匹配的自动方法认为匹配的词是与源语言文本相关的，而没有匹配到的词是与源语言文本不相关的，是没有意义的。一般而言，对于一个给定的源语言文本，往往有许多可行的翻译，而这些翻译在选择词上可能不尽相同。比如，对汉语词“军队”，英语可以翻译成“army”，“military”，“troop”等，如何选用与具体语境相关。因此，词语翻译的不同可能不能算作翻译上的错误。虽然可以通过引入多参考译文来在一定程度上缓解这个问题，仍然可能有许多有意义的词被视作未匹配词，这样就会对系统输出译文做出不公正的评价。换言之，在未匹配词中，有些词确实是翻译错误的，而有些词是翻译正确的，只是选择的词和参考译文不一致。忽略这些有意义的未匹配词可能会限制基于N-gram的自动评测方法[5]。

本文试图通过将模糊匹配引入机器翻译自动评测来解决上述问题，基本思想是词与词之间的匹配程度应当是一个介于 0 和 1 之间的一个实数，而不是非 0 即 1。在第 2 节，我们将详细介绍模糊匹配的原理，并且通过一个例子来说明如何搜索模糊匹配的词对以及计算该词对的相似度。实验结果和分析在第 3 节给出。最后一节是结论和我们将来的工作。

2. 模糊匹配

2.1 相似度

在机器翻译自动评测中，匹配是一个基本的操作。所谓匹配，就是考察系统输出译文的一个词和参考译文中的一个词是否相同。我们用相似度来衡量这种匹配程度。传统上，如果两个词相同，则相似度为 1，否则为 0。比如，“army”和“army”之间的相似度为 1，而“army”和“military”之间的相似度为 0。然而，“army”和“military”是在意义上是相近的。如果对源语言文本中的一个词“军队”，系统输出译文翻译成“military”，而参考译文翻译成“army”，我们应当认为“military”是一个很好的翻译。

因此，词与词之间的匹配程度应当是一个介于 0 和 1 之间的一个实数，而不是非 0 即 1：

$$similarity(word_1, word_2) \in [0,1] \qquad \text{公式 1}$$

我们定义将系统输出译文中的词分为三个类，如表 1 所示：

相似度	类别
0	未匹配词
(0, 1)	模糊匹配词
1	完全匹配词

表 1：未匹配词、模糊匹配词和完全匹配词

2.2 图形化表示

(Turian et al, 2003)提出了一种基于unigram计算F-measure的机器翻译自动评测方法GTM（General Text Matcher）[6]。GTM采用自然语言领域常用的准确率、召回率和F1 值来衡量机器翻译系统输出译文的质量。GTM将系统输出译文和参考译文之间的匹配用坐标系来表示。下面，我们将用一个例子来说明。需要注意的是，我们的图和(Turian et al, 2003)不完全相同。例子中的句子选自(Papineni et al, 2001)：

例子:

源语言文本：党指挥枪是党的行动指南。

参考译文：*It is a guide to action that ensures that military will forever heed party commands.*

系统输出译文：*It is insure the troops forever hearing the activity guidebook that party direct.*

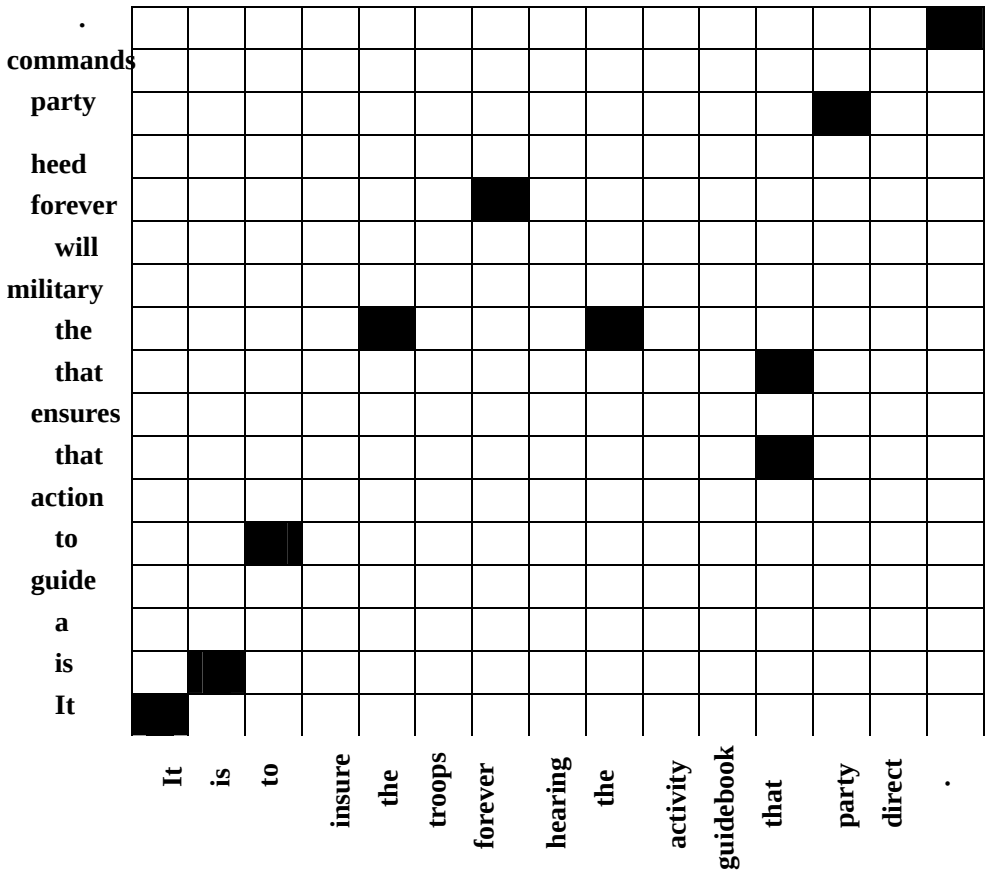

图 1：完全匹配

如图 1 所示，系统输出译文的词在 x 轴，参考译文的在 y 轴。每个词通过它在句子中的位置获得其坐标。例如，“insure”是系统输出译文的第 4 个词，因此其坐标是(4, 0)。当图中某个点横坐标和纵坐标上对应的词相同，我们将其标为“■”，并称之为一个点。一个点对应着一个词对，如点(3, 5)对应着词对{to, to}。我们称图 1 为匹配图。

如果系统输出译文和参考译文有连续的词对匹配，那么在匹配图中则表现为连续的、平行于 y=x 的点序列，(Turian et al, 2003)称这样的点序列为 run。在图 1 中，对应于“*it is*”的点序列是一个 run。

如果某些点出现在同一行或者同一列，我们称这些点是相互冲突的。通常冲突的点是虚词。对于相互冲突的点，我们只保留其中的一个点，删除其余的点。在图 1 中，点(5, 10)和点(9, 10)是相互冲突的。

2.3 LCCSR

前文提到，一个词对的相似度应当是介于 0 和 1 之间的一个实数。我们如何来计算相似度呢？很自然地，我们会想到利用现有的语言资源，比如 WordNet。事实上，利用 WordNet 来计算词语的相似度是很有意义的。然而，在本文我们试图独立于语言资源来自动搜索模糊匹配的词对和计算相似度，因为这样的方法更具备一般性。将来，我们会利用语言资源来改善我们的方法。在本文中，我们讨论的是目标语言是英语的情况。

一般而言，两类词对应当具有较高的相似度：近义词和同根词。例如，{army}和{military}是近义词，{absent}和{absence}是同根词。

对于同根词，我们假设两个词共有越多的连续的字符则相似度越高。我们提出用LCCSR（Longest Continuous Common String Ratio）来计算同根词的相似度。LCCSR类似于(Melamed1995)提出的LCSR（Longest Common String Ratio）[7]。LCSR是用来计算计算两种语言的词之间的同根度（cognate-ness）。我们用LCCSR而不是LCSR是因为我们认为连续的共有子串对于计算相似度是重要的，而LCSR不强调这种连续性。LCCSR的计算方法如下：

$$LCCSR(w_1, w_2) = \frac{|LCCS|}{\max\{|w_1|, |w_2|\}} \qquad \text{公式 2}$$

其中，w_1 和 w_2 是词对的两个词，LCCS 是最长的连续共有子串，|str|表示字符串 str 所包含的字符数。

我们假设“两个词共有越多的连续的字符则相似度越高”，这一假设并不总是成立。比如“be”和“bee”的 LCCSR 是 0.67，但是这两个词却是无关的。这个问题可以通过引入语言资源来解决，目前我们通过加入以下限制来缓解这个问题：

限制 1：计算 LCCSR 只在实词之间进行

限制 2：w_1 和 w_2 的长度必须均大于 3

LCCSR 对于计算同根词比较有效，但是对于近义词却不很有效。比如，{army, military}的 LCCSR 是 0.25。我们在下小节解决这个问题。

2.4 模糊匹配策略的原理

我们称语义上相关但是字面上不相同的词对为模糊匹配的词对，其在匹配图中对应的点

为模糊匹配点。例如，{army, military}是一个模糊匹配的词对。

如何来自动地搜索这些模糊匹配的词对？我们的策略是首先假设一些候选的词对，然后再从中筛选。我们把词分为两类：虚词和实词。由于虚词的数目是有限的，通过建立一个虚词表就很容易地区分虚词和实词。我们在所有实词之间做比较，假设所有的实词对都是可能模糊匹配到的。这样，我们就得到许多候选词对。将这些候选词对对应的点加入到匹配图中。此后，我们运用一个删点算法来对候选点进行过滤，剩下的点就是模糊匹配的点。最后，为每个词对计算相似度。

整个过程可以由六个步骤来实现，执行的中间过程见图 2：

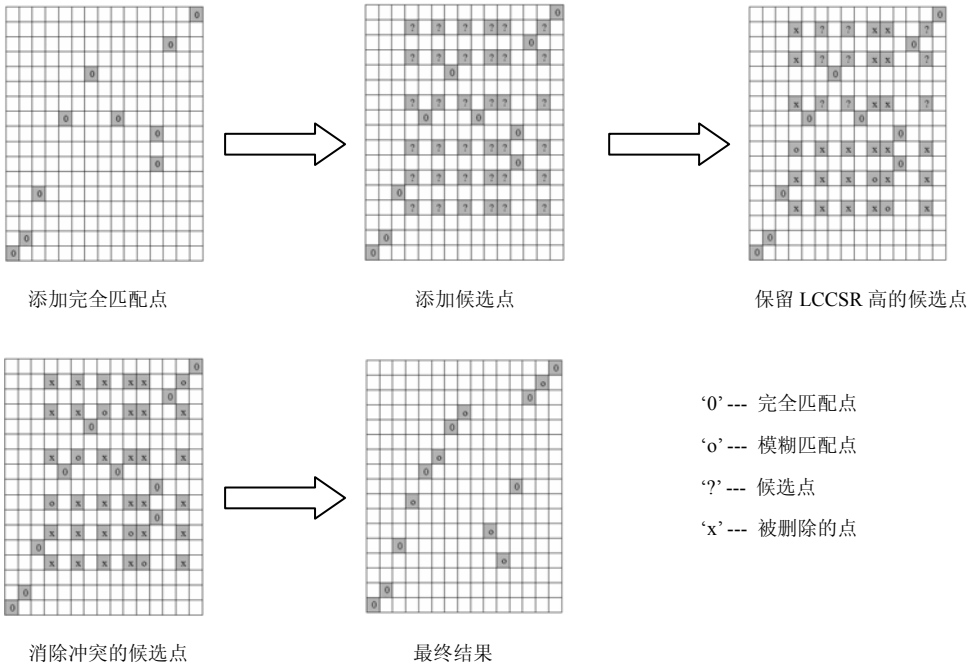

图 2：搜索模糊匹配点的过程

第 1 步：添加完全匹配点

考察系统输出译文和参考译文，找出完全匹配的词对，将完全匹配点加入到匹配图中。

第 2 步：第一次删除完全匹配点

我们为每个点赋予一个属性：runLen。runLen 是每个点所在 run 长度。如果一个点是孤立的，则其 runLen 为 1。保留冲突点集中 runLen 最大的点，删除其余的点。如果冲突点集中所有点的 runLen 都相等，则不删除任何点，留到后面（第 5 步）处理。

第 3 步：添加候选点

将未匹配词分为虚词和实词，将实词之间所有可能的匹配点加入到匹配图中。

第 4 步：删除冲突的候选点

通过研究大量的词对，我们发现 LCCSR 越高词对越相似。但是当 LCCSR 较低的时候，LCCSR 便不能较好地反映相似度。因此，我们将 LCCSR 不低于 0.5 的候选点设为模糊匹配点，并删除与之冲突的候选点。对于 LCCSR 低于 0.5 的候选点，若不与模糊匹配点冲突，

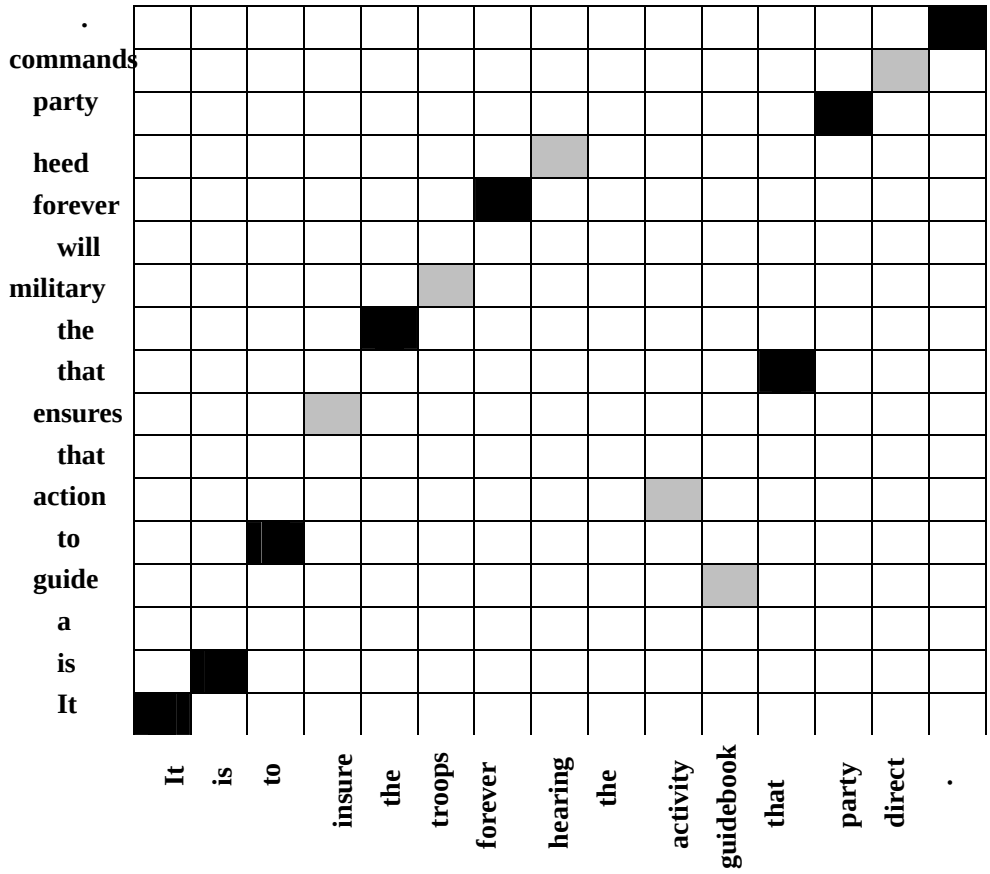

图 3：模糊匹配图

则仍保留，如点(6,11)。经此处理后，仍将有冲突的点存在。一般而言，一个句子往往是通过几个概念连接而成的，而为了表达一个概念，往往需要几个词连在一起。设一个点的坐标是(x, y)，则定义它的邻居为(x-1, y-1)和(x+1, y+1)。若一个点是匹配点（完全匹配或者模糊匹配），则它的邻居很有可能是模糊匹配点，因为邻居很有可能是与匹配点一起表达某个概念。匹配点的 runLen 越大，则其邻居越可能是模糊匹配点。我们引入连接度来表示候选点的这种性质，所谓连接度就是假设候选点是模糊匹配点，形成的新 run 的长度。对于冲突的候选点，我们优先保留连接度大的点，删除与之冲突的候选点。

第 5 步：第二次删除完全匹配点

由于加入了模糊匹配点，第 2 步不能判断的冲突的完全匹配点的 runLen 可能发生变化，这时候再保留 runLen 大的点。如果 runLen 没有变化，则保留距离 y=x 最小的点。

至此，我们得到了一个包含完全匹配点和模糊匹配点的匹配图，如图 3 所示，其中黑色的方格表示完全匹配点，灰色的方格表示模糊匹配点。

第 6 步：计算相似度

根据模糊匹配的最终结果，我们便可以为每个点计算相似度。显而易见，完全匹配的点的相似度为 1，所以我们集中解决模糊匹配点的相似度计算。

模糊匹配点的相似度取决于两个因素：一个是字面相似度 LS(Literal Similarity)，一个是结构相似度 SS(Structural Similarity)。相似度计算如下

$$LS = \begin{cases} LCCSR & LCCSR \geq 0.5 \\ 0 & LCCSR < 0.5 \end{cases}$$ 公式 3

$SS = confidence * ratio_{run}$　公式 4

结构相似度并不可靠，特别是当完全匹配点比较少的时候。根据马太效应，即“贫者愈贫，富者愈富”，我们做出这样的假定：完全匹配点越多，则翻译者的翻译水平可能更高，所找到的未匹配点越可能是可靠的。我们定义可靠性的度量 confidence，利用 conference 来调节 SS，其计算公式为：

$$confidence = \frac{2 * count_{full-matched}}{count_{ref} + count_{can}}$$　公式 5

其中，$count_{full-matched}$ 是完全匹配点的数目（去掉冲突点），$count_{ref}$ 是参考译文中词语的数目，$count_{can}$ 是系统输出译文中词语的数目。

在图 3 中，$count_{full-matched}$ 为 8，$count_{ref}$ 为 17，$count_{can}$ 为 15，则 confidence 为 0.5。

$ratio_{run}$ 的计算公式如下：

$$ratio_{run} = \frac{runLen}{runLen_{\max}}$$　公式 6

其中 runLen 是该点的 runLen，$runLen_{\max}$ 是匹配图中最大的 runLen。

在图 3 中，点(6,11)的 runLen 长度为 2，匹配图中最大的 runLen 为 3，则点(6,11)的 $ratio_{run}$ 为 0.67。

如何由LS和SS获得相似度，我采用了[Tiedemann 2003]提出的对各种线索进行析取的方法[8]。最终的相似度计算公式为：

$similarity = LS + SS - LS * SS$　公式 7

表 2 列出图 3 中模糊匹配点的相似度。

点	词对	相似度
(4, 8)	{insure, ensures}	0.7619
(6, 11)	{troops, military}	0.3333
(8,14)	{hearing, heed}	0.3333
(10, 6)	{activity, action}	0.5833
(11, 4)	{guidebook, guide}	0.6296
(14, 16)	{direct, commands}	0.5

表 2：模糊匹配点的相似度

3 实验结果及分析

我们设计了两个实验来考察模糊匹配在实际中的应用。

第一个实验是考察模糊匹配是否能够很好地找出有意义的未匹配的词对。选择 100 个中文句子，由翻译人员翻译一份参考译文，然后由一个机器翻译系统翻译系统输出译文。100 个句子中以短句子居多，参考译文平均每个句子包含 11.795 个词，系统输出译文平均每个句子包含 11.925 个词。我使用准确率、召回率和 F1 值来衡量模糊匹配的预测能力，结果如下：

准确度	召回率	F1 值
79.33%	78.81%	79.07%

表 3：准确率、召回率和 F1 值

通过研究，我们发现大多数的匹配错误发生在以结构相似度为主计算相似度时，也就是 LS 为 0 时。前面已经提到了，结构相似度并不是非常准确，特别是完全匹配点比较少的时候。目前的解决方法是，对于结构相似度起主导作用的点赋给一个较低的相似度。

第二个实验是考察模糊匹配是否能够提高现有自动方法的性能。我将模糊匹配加入到 BLEU 中，和完全匹配的 BLEU 进行比较。语料是 863 机器翻译自动评测汉译英的对话语料。

主要通过修改计数来引入模糊匹配。例如，当比较两个 N-gram：$w_1 w_2 ... w_n$ 和 $w'_1 w'_2 ... w'_n$ 时，其相似度为 $\min\{similarity(w_i, w'_i)\}$。

实验结果如表 4 所示：

	人工评分	BLEU(完全匹配)	BLEU（模糊匹配）
S1	0.6158	0.1855	0.3410
S2	0.4384	0.1702	0.3190
S3	0.4463	0.1225	0.2875
S4	0.7316	0.3893	0.4998
S5	0.5039	0.1685	0.3218
S		0.7000	0.9000
P		0.8862	0.9036

表 4：完全匹配的 BLEU 和模糊匹配的 BLEU

其中，S1，S2，S3，S4 和 S5 分别是 5 个机器翻译系统的编号。S 是 Spearman 相关系数，P 是 Pearson 相关系数。

从表 4 可以看出，加入模糊匹配后，BLEU 的评分明显增高。Pearson 相关系数略微提高，Spearman 相关系数提高很多，这是因为模糊匹配纠正了完全匹配的一个排序错误：人工评分认为 S5 比 S2 的译文质量高，基于完全匹配的 BLEU 则认为 S2 比 S5 的译文质量高，基于模糊匹配的 BLEU 则纠正了这个错误。由此也可以看出，模糊匹配可能会提高自动方法的“认知”精度的，可以用于提高其他的自动评测方法。

4 总结

实验表明，通过引入模糊匹配能够较好地搜索到有意义的未匹配点。更重要的是，通过引入模糊匹配，BLEU 的性能得到了提高。模糊匹配可以用来提高其他的自动评测方法。

尽管将模糊匹配引入机器翻译自动评测很有意义，仍然有一些问题需要解决。我们认为搜索模糊匹配点和计算相似度是两个关键问题，需要更深入地研究。对于当前方法搜索模糊匹配点的能力，需要持谨慎态度，因为当句子更长更复杂时可能搜索到的会是没有意义的词对。可以通过引入新的特征来改善这个问题，目前我们是通过降低相似度来克服这种负面效应。本文研究的是对英语的处理，可以扩充到其他语种。我们将采用如 WordNet 和 HowNet 等语言资源来提高现有方法。

参考文献

[1] Kishore Papineni et al. BLEU: a Method for Automatic Evaluation of Machine Translation Evaluation [A]. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311--318, Philadelphia, Pennsylvania, July 2002.

[2] 俞士汶，姜新，朱学锋。机器翻译译文质量评价的实践与分析[A]。中文电脑国际会议 ICCC'94（新加坡）论文集，PP26-32。

[3] 罗爱荣，段慧明。机译评估方法评述及一个基于测试集的自动评估系统——MTE 的进展[J]。摘要发表于《计算语言学进展与应用》，陈力为、袁琦主编，清华大学出版社，1995。

[4] George Doddington. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics [A]. In Human Language Technology: Notebook Proceedings [C]: pp128-132. 2002.

[5] Christopher Culy and Susanne Z. Riehemann. The Limits of N-Gram Translation Evaluation Metrics [A]. In Proceedings of Machine Translation Summit IX Workshop "Machine Translation for Semitic Languages: Issues and Approaches" [C], New Orleans, USA, 23-28 September 2003.

[6] Joseph Turian, Luke Shen, and I. Dan Melamed. Evaluation of Machine Translation and its Evaluation [A]. In Proceedings of Machine Translation Summit IX Workshop "Machine Translation for Semitic Languages: Issues and Approaches" [C], New Orleans, USA, 23-28 September 2003.

[7] I. Dan Melamed. Automatic Evaluation and Uniform Filter Cascades for Inducing N-best Translation Lexicons [A]. In Proceedings of the Third Workshop on Very Large Corpora [C]. Boston, MA, 1995.

[8] Jörg Tiedemann. Combing Clues for Word Alignment [A]. In Proceedings of the 10th Conference of the European Chapter of the ACL (EACL03) [C], Budapest, Hungary, April 12-17, 2003.

附录：搜索模糊匹配点的详细说明

本附录是对正文 2.4 节的详细说明。

第 1 步：添加完全匹配点

向匹配图中添加完全匹配点图 4 所示，完全匹配点用“0”表示。

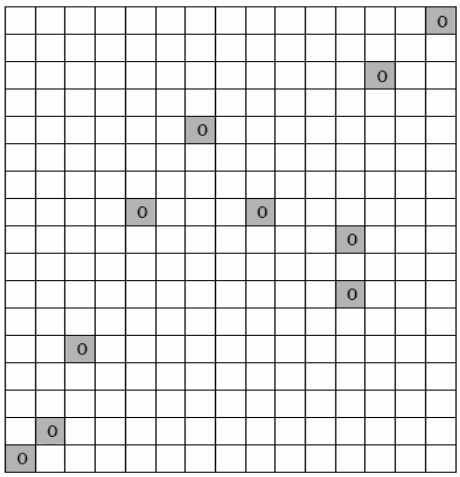

图 4：添加完全匹配点

第 2 步：第一次删除完全匹配点

由于点(5, 10)和(9, 10)，点(12, 7)和(12, 9)的 runLen 相同，不做改变。

第 3 步：添加候选点

我们将参考译文和系统输出译文中的未匹配词分类如表 5：

参考译文	虚词	a, will
	实词	guide, action, ensures, military, heed, commands
系统输出译文	虚词	
	实词	insure, troops, hearing, activity, guidebook, direct

表 5：参考译文和系统输出译文中未匹配词的分类

在参考译文和系统输出译文的实词之间构造候选点，加入匹配图，如图 5 所示，候选点用“?”表示：

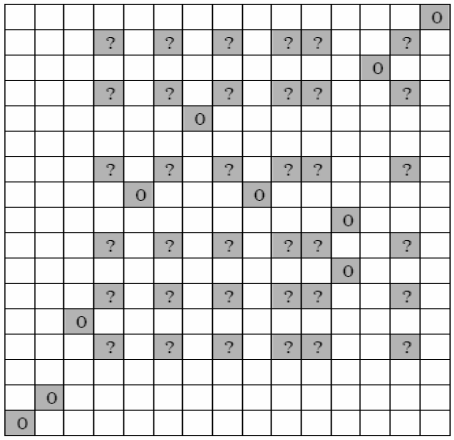

图 5：加入候选点

第 4 步：删除冲突的候选点

首先计算候选点的 LCCSR，然后选择 LCCSR 不低于 0.5 的候选点，如表 6 所示：

点	词对	LCCSR
(4, 8)	{insure, ensures}	0.7143
(11, 4)	{guidebook, guide}	0.5556
(10, 6)	{activity, action}	0.5

表 6：LCCSR 不低于 0.5 的候选点

将 LCCSR 不低于 0.5 的候选点设为模糊匹配点，删除与之冲突的点。对于 LCCSR 低于 0.5 的点，只要不与模糊匹配点冲突，便保留。结果如图 6 所示，模糊匹配点用“o”表示，被删除的点用“x”表示：

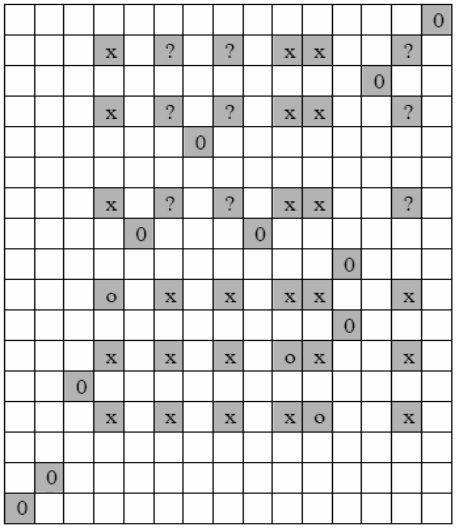

图 6：将 LCCSR 不低于 0.5 的候选点设为模糊匹配点，并删除与之冲突的候选点

图 6 中仍有一些冲突点未处理，我们计算这些冲突的候选点的连接度，结果如表 7 所示：

候选点	连接度
(14, 16)	3
(8, 14)	2
(6, 11)	2
(8, 11)	1
(14, 11)	1
(6, 14)	1
(14, 14)	1
(6, 16)	1
(8, 16)	1

表 7：候选点的连接度

之后，将连接度大的候选点设为模糊匹配点，删除与之冲突的候选点。删除所有连接度为 1 的候选点。处理后的结果如图 7 所示：

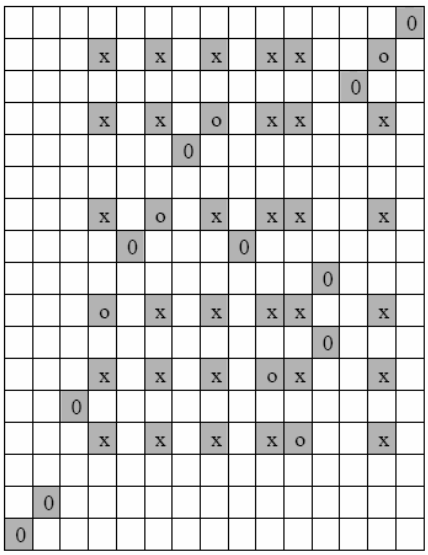

图 7：消除冲突的候选点

第 5 步：第二次删除完全匹配点

在图 7 中，仍有冲突的完全匹配点：点(5, 10)和(9, 10)，点(12, 7)和(12, 9)。由于点(5, 10)的 runLen 已经变成 2，因此删除点(9, 10)。对于点(12, 7)和(12, 9)，保留离 y=x 最近的(12, 9)。

至此，所有的冲突均消除完毕，最终结果如图 8 所示：

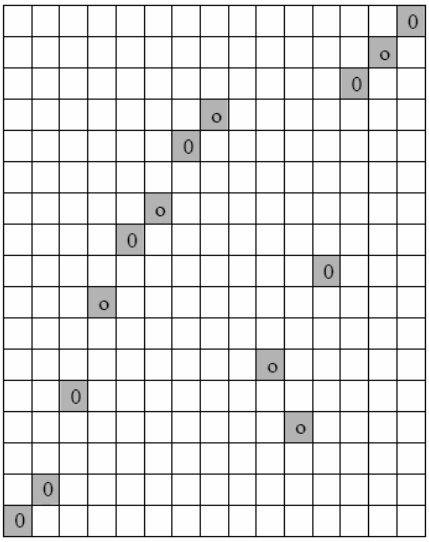

图 8：模糊匹配图