1. **2011-****Automating String Processing in Spreadsheets Using Input-Output Examples**

最基础的原型。

算法处理I/O paris的思路：**To extract substrings, FlashFill runs LCS on every input-output string pair in the I/O set to get a set of substring candidates. It then takes the entire set of substring candidates and simply tries every possible regex and constant index that can be used at substring boundaries**

缺陷：

The algorithm presented in that work assumes strings to be a sequence of characters and can only perform concatenation of input substrings and constant strings to generate the desired output string.[源自2012数字提取的**七、相关工作**]

1. 2011-Learning Semantic String Transformations from Examples 添加了多表搜索的能力
2. **2012-Spreadsheet Data Manipulation Using Examples** 对上面两篇文章的总结与归纳(一定要和上面的结合起来看)。
3. **2012-Synthesizing Number Transformations from Input-Output Examples** 实习生作品，在2011原型的基础上，加入的处理数字的能力(加入了几条小公式)。**重点学习对象**，学习如何小幅改进模型！**又因为他引入了处理数字(大于、小于某个值等)，所以可以用于数据清洗！**
4. **A Colorful Approach to Text Processing by Example.** ：引入了带**结构化本文的处理能力**(比如一本书有署名、作者、出版时间、价格等)。但是文章说的不清楚，暂时没挖掘出利用价值。不过有很多引用了此文章的后续文章出现。
5. (colorful的进阶,不属于数据提取)**2016 Program Synthesis using Natural Language** ——colorful带来了更复杂的操作，可以通过NLP，输入需求，识别需求，完成输出。
6. 2013-A Machine Learning Framework for Programming by Example 从文本特征中学习得到rules，clues的权重，以加速问题求解。

注：不再使用2011年原型中所用的A\*方法搜索解空间，而是通过学习本文特征，根据概率来决定要使用哪种小程序。(**本文着重处理多行input的问题，对于单行输入，使用原型方法更为有效，也就是说本文没有从本质上提升程序性能。 本文中心在加速，可不看**)

1. 从txt或web提取数据 **[PLDI 2014]: FlashExtract: A Framework for Data Extraction by Example**s ：手动选择某个region，程序会找出所有符合这个region的数据；如果用户对结果不满意，还可以添加positive examples或negative examples。

与前面的工作不同，本文定义的主要是几个**operation(Map,merge，pair等)**，可以把这个模型应用到任何DSL中。

功能很强，可以在<https://prose-playground.cloudapp.net> 进行测试。提取得到数据后，还可以接着采用substring、number等处理方式进一步处理，是”自动数据清洗”的基础。

虽然本文对算法过程解释的较详细，但是仍然比较难理解，尤其是他的几个定义(**sequence or struct-ancestor**、**region&field**、colorful region、**一致性和完整性**等)；各个算法**何时调用**好像也没有说清楚，每个算法中含大量未知符号，难以理解。

本文需要反复阅读加以揣摩。争取2月以内能弯成一个原型(没有GUI也行)。

1. [oopsla15-pbe]**FlashMeta a framework for inductive program synthesi**s 感觉像对过去近10个工作的总结，如果要做2011或flash extract的话就需要看这个。
2. **Bias reformulation for one-shot function induction** 算对2011的改进(称2011也是一种one-shot-learning)，他的abstract和introduction适合翻译，适合当毕设论文的开头。

将原本用于关系预测(relational predicates)的ILP(inductive logic programming)应用到函数预测上(functional predicates)。使用最近发展起来的Meta-Interpretive Learning framework，根据positive和negative examples去进行逻辑推演预测。

**但是本文有一点奇怪，像是水文，可以略过不看。**

1. **NEURO -SYMBOLIC P ROGRAM S YNTHESIS(总结了2011原型，用DL改进了原型。在完成原型之后优先考虑实现此论文)**

incrementally expanding partial programs,R3NN employs a tree-based neural architecture that sequentially constructs a parse tree by selecting which non-terminal symbol to expand using rules from a context-free grammar。

用二叉树表示程序，中间节点为rule(不懂)，叶子节点为符号(终结符和非终结符)。

本文重点：

1. 在拓展程序(expand PPT)时选用个部件的概率以及如何评价这次拓展(ranking)
2. 通过2个双向LSTM作为第一层encoder，再把第一层的2个output作为另一个encoder的input，用于产生condition。

Substring：不再像FlashFill那样用LCS+正则匹配人工处理，而是用机器学习自动处理：

**To extract substrings, FlashFill runs LCS on every input-output string pair in the I/O set to get a set of substring candidates. It then takes the entire set of substring candidates and simply tries every possible regex and constant index that can be used at substring boundaries.**