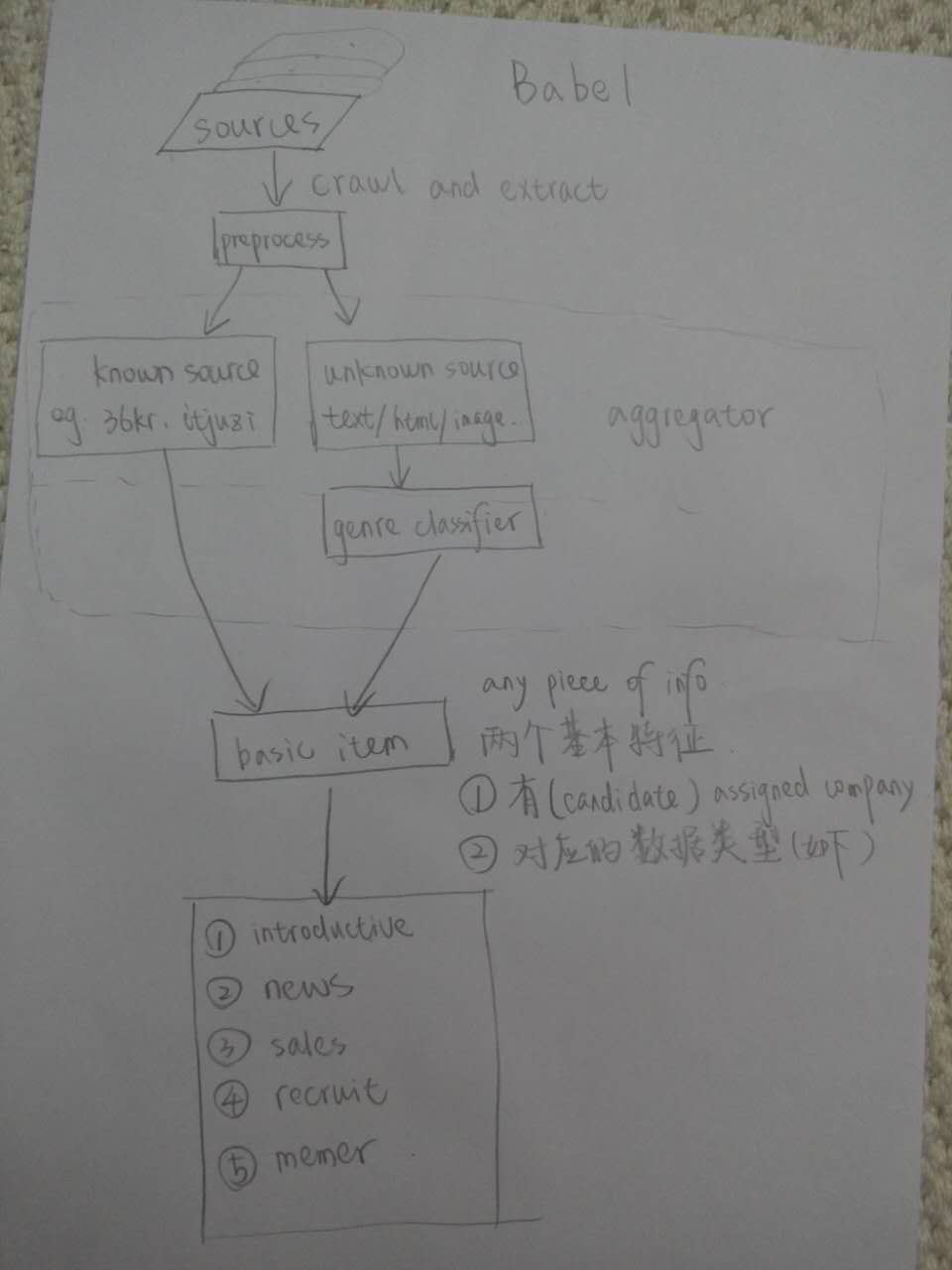
# 三个维度

系统的智能部分由三个维度来定义。

第一个维度是数据收集和整合，着重的是如何将互联网中每一条信息收集起来，并结构化或者半结构化地对应到系统的数据架构，也就是如何用我们业务需要的方式来组织互联网中的（全部）有用信息。

第二个维度是检索，着重的是用户如何主动获取信息。

第三个维度是发现，着重的是系统和用户的互动，系统主动给用户提供有效信息。



# 数据收集和整合(Code: Babel)

## 架构

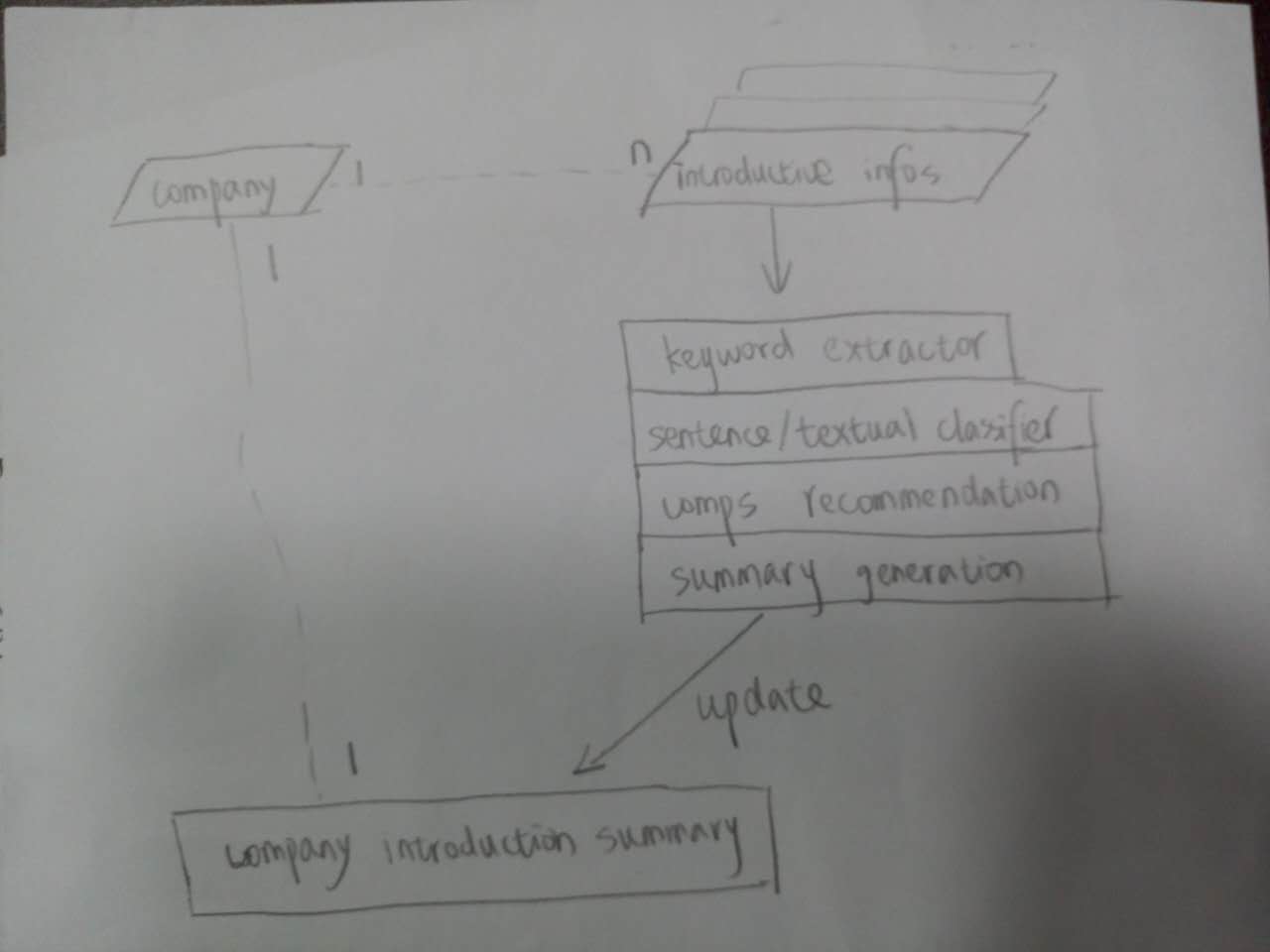
假设系统完善的情况下，关于一个创业公司，互联网中存在的关于它的有价值的信息都应该被收录和整合进来。爬虫模块解决了给定信息源的获取问题，这个模块意图解决的是对信息进行有效的整合。

首先，对于互联网中每一个有URI标识的信息，称每一条为一个sourceURI，我们用一定的策略捕捉到其中的一些（最佳的策略就是尽可能多地捕获与创业相关的source，现在的策略是从特定的源来捕捉这些source，搜索引擎的策略是捕捉全部的source），使用爬虫进行抓取，并提取，这部分对应现有的spider和parser。

第二步进行预处理，针对给定的源比如36kr，和现在的方法类似，由于已知网页的结构，可以提取相关信息。对于未知的源，也就是对应的互联网上任何一条信息，形态可以是文本，网页或者图片等等，进行分类处理和关键信息提取。不管是前者还是后者，都要生成一个系统的basic item，这个基本数据可以是从任何源来的，必须包含两部分信息，第一是对应到的（可能的）公司；第二是分类结构，也就是Babel接下来要如何看待这个basic item。这部分对应现有的aggregator。

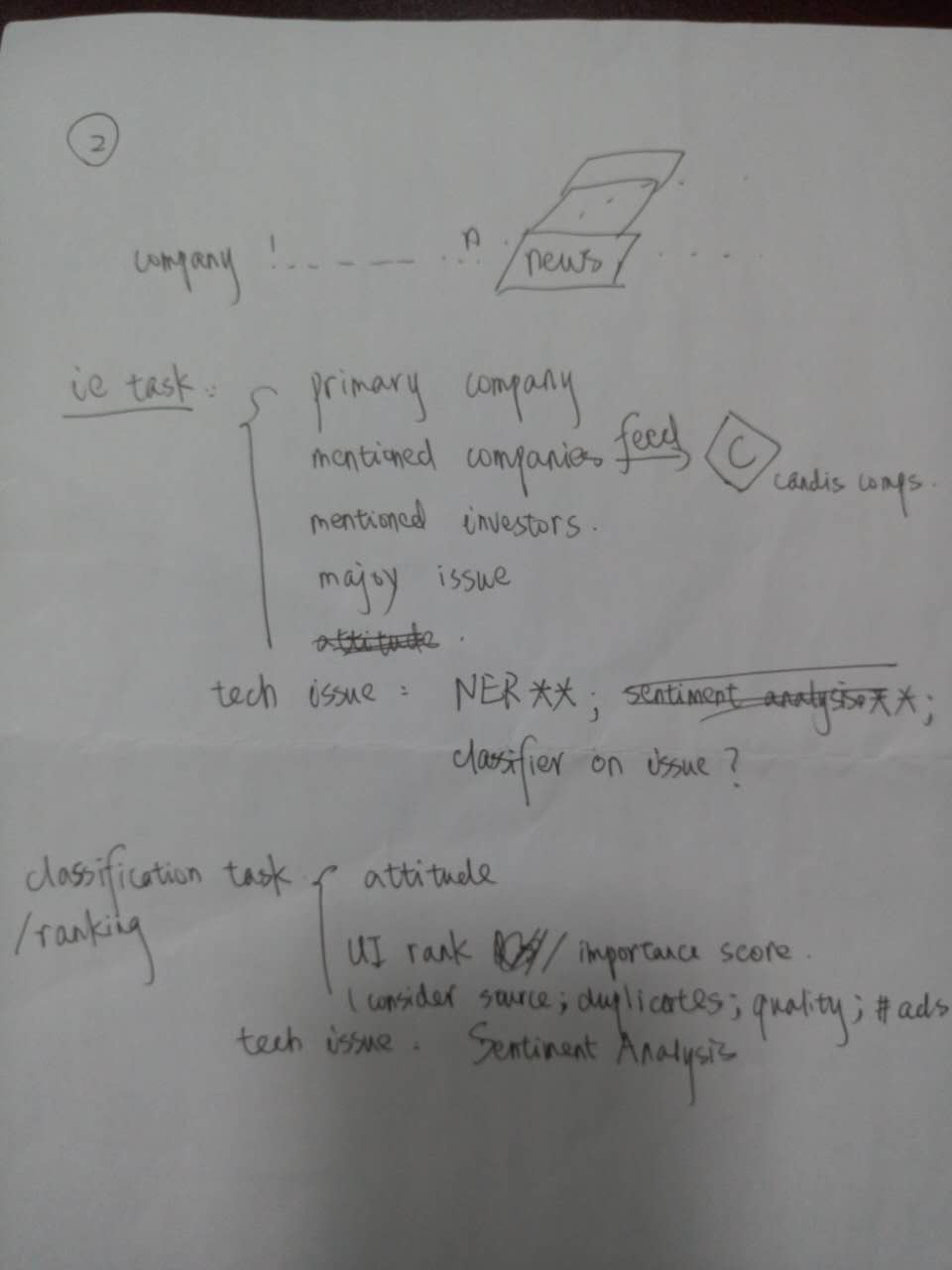
最后一步是信息的整合添加。每一条basic item进入对应类型的处理模块，我现在定义了五种类型：

1. introductive，公司的介绍性信息，包括介绍，所在地，融资信息等等。



1. news，新闻。

新闻部分我还没太想透，想到的主要是一些信息抽取的情景，比如给定一篇未知科技新闻，抽取出新闻中的关键公司，提及公司，提到的投资者/投资人，新闻的事件类型分类，新闻对提及事件/人物的评价/态度，etc。新闻相关的东西看似比较简单，但是需要更逻辑化地梳理一下。



1. sales，销售信息，针对一些有商品出售的公司适用，包括销售情况和用户在电商等渠道对产品的反馈分析/App在应用市场的评论分析。

这个部分思路比较清晰简单，实现上可能需要稍微多一点儿的man hour来训练和调通。主要的技术细节有两点，关于用户-评分的fraud analysis和关于用户-评论的sentiment analysis，也就是分析电商商品/应用App是不是刷单，尽量给出用户真实的反馈评价。类似对公司产品的用户访谈挖掘。

1. recruit，招聘信息，包括招聘情况评价等。

这部分最直观的就是之前工作的加深。自动生成关于招聘信息的报告，评价公司的招聘规模、地域，比较同类公司的招聘待遇，估算公司的招聘速度。

1. member，成员信息，包括成员的经历评价和社交网络，社交关系等等。

通过(社交)网络信息给团队成员进行标签化，便于快速理解。主要作用是让人对创业者有一个快速的判断，不过这个判断的意义是否很大？

这部分可能对数据抓取的要求比较大，对应和去重可能也得仔细考虑。

## 技术要点

Babel主要要处理的是非结构化的文本，对应到系统的结构。

### 分词/实体识别

实体识别(Name Entity Recognition, NER)是建立在分词的基础之上的，目的是识别出文本中的实体，主要包括公司/产品名，人名和投资机构名，进而识别出文本中的事件以及将文本对应到公司。

短期的思路是在开源分词方案jieba的基础上，通过已知的结构化数据源信息（比如从36kr等了解到的公司名人名等）进行简单的训练，长期上可以通过自主标注语料来获得更精确的模型，不过暂时必要性不是很大。

### 分类

系统中涉及到多种分类(Classification)，比如文本的类型分类，公司的行业分类，等等，每个具体分类问题的流程基本相似，分类器都要特殊训练。

基本流程为 特征提取—特征选择—验证

### 基于内容的推荐

相似公司的推荐基本还是基于内容的，因为浏览记录/用户量不那么大，不同于电商类的协同过滤。两个思路来处理这个问题，

1. 文档表示—文本相似度计算。现行方案用简单的词向量和tf-idf，辅以简单的降维，可以尝试更多种降维方法，和现在比较流行的基于神经网络的word embedding思路的表示方法，后者可能会比较适合现在一个公司对应多个文本的情况。
2. 基于图的文档表示—图连通问题。可以通过tag/投资机构/领域/创业者背景等因素作为节点，将公司作为要连接的节点，把这个问题当做一个网络分析问题。

### 关键词提取

关键词提取和上面提到的文档表示有一定的关系，我们可以这样理解这个问题：

原始文本（信息最全，但是分散）----短的文本总结（信息丰富集中）----几个关键词（信息很集中）----子行业分类（信息少，精确）----行业分类（信息更少，笼统）。从信息论的角度来看，这几个问题是相通的，相当于把信息一点一点压缩，每一个环节会损失一些信息，信息量更小。

现在关键词提取的解决方案，分词-加权的textrank，结合word2vec。问题是textrank比较容易提取出来非常见词（因为原始文本质量不统一），embedding使用不太够，多文本提取方案不成熟。

可以尝试的几个方向

1. 更多使用embedding，尝试一下现在比较火的深度学习方法，直接的文档表示没看到比较好的方法，词向量的效果应该还可以
2. 更多使用已知源的tag，训练一下当成一个分类问题

# 检索 (Code: Heart of Gold)

## 启发式搜索

## 语义搜索

## 结果展现方式

# 发现/推荐 (Code: M. Daniel)

## 个性化项目推荐

## 个性化分析报告，eg. 行业趋势；关键词变化历程