This lecture gives an overview of three important NLP applications. It's a bit unorthodox to cover this *now*, as opposed to much later in the semester, but we think a high-altitude view of some of the ends will help you to understand the means, and also help you get started with your project.

Information Extraction (IE). IE is essentially the problem of transforming unstructured text into relational databases (or other structured knowledge bases). The subproblems include **named entity recognition**, reference resolution (which mentions actually refer to the same entity?), **entity relation recognition**, event recognition, and temporal analysis . We'll come back to some of these topics; today we discuss $\{\text{named entity, relation}\} \times \{\text{detection, classification}\}$. Named entities: ambiguity makes this hard. A standard representation is `B-I-O` labeling. Most commonly we think of detection and classification as a single task, carried out using supervised learning. Evaluation by **precision** and **recall**.

Entity relations: after finding named entities, we can treat this as two classification problems (first detect, then classify), then evaluate against gold-standard data by precision and recall. Alternate approach is **bootstrapping**. Start with some seed patterns, find examples of relations, search for those relations, find new patterns, repeat. Evaluation: precision needs humans; recall is much harder—how do you know if you found everything in a huge corpus?

Information Retrieval (IR). (11-441 is a special topics course about IR, offered in the fall.) Given a query, recover a ranked list of documents. The goal is to get relevant documents to the top of the list. The **vector-space model** is very widely used. Essentially, every document and every query are represented as real-number vectors, with one coordinate per word type. The weights are selected in various ways; *tf-idf* works pretty well:

$$x_w = (\#\text{ of times } w \text{ occurs in } x) \times \log \frac{\#\text{ of docs total}}{\#\text{ of docs containing } w}$$

The **cosine similarity** (normalized dot-product) gives a score to each document, for the query. There are many twists! Evaluation is more nuanced than just precision and recall, because there's a tradeoff. Precision-recall curves tell more of the story (see text for some simpler summary scores you can derive from such a curve).

Question Answering (QA). Usually the focus is on **factoid** questions (with simple answers). QA systems are usually built out of many smaller components (such as those above and others coming later in the course). The major steps: turn a question into a query for an IR system, predict the type of the answer, select the most useful passages from retrieved documents and rank them, select the right sequence of words for the answer, and assemble. Commonly we evaluate using **mean reciprocal rank**, i.e., the average over test questions of the inverse rank of the first correct answer that the system provides.