Scientific Communication

David Chiang and Kevin Knight
Spring 2014

0 Introduction

The goal of this workshop is to develop scientific and technical communication skills. It is mainly about getting your ideas across effectively. It is not about English grammar, English style, mathematical writing, or TeX, except maybe for some tips here and there.

Why is communicating your work important? First, it enriches your interaction with your colleagues. If you communicate your work well to them, they will be able to give you better feedback and will want to form collaborations with you.

Second, your interactions with colleagues will create a better impression on them. They will spread the word about your work to others. Good communication is critical for getting a job and succeeding in it.

Third, it will help you think better about your own research. Writing up your work forces you to think critically and logically about it, and ultimately will make it better.

Overview

Section 1 will be about general principles. Then we will proceed through the main parts of a paper or talk:

2. How to talk about the goal of your work
3. How to talk about your method: concepts
4. How to talk about your method: procedures
5. How to talk about your results
6. How to talk about other people's work

We will also talk about particular communication skills:

- Research proposals
- The elevator pitch (part of session 2)
- Papers
- Presentations
- Posters

During each session, we will discuss the homework from the previous section, present a handout on a new topic, and assign some homework for next time.

Further reading

Simon Peyton-Jones at Microsoft maintains a good page on technical communication, including slides from two of his own talks (http://bit.ly/17kX4l). The scope of these talks is similar to this workshop.

The book that is widely considered the Bible of writing is Strunk and White's *Elements of Style*. We don't recommend it. A much better book is: Joseph M Williams, *Style: Lessons in Clarity and Grace*. This book focuses on the low-level details of writing good sentences, so is complementary to this workshop. David has a copy that you can borrow.

For help with English, maybe: Swan, *Practical English Usage*. Or, more systematic and technical: Huddleston and Pullum, *A Student's Introduction to English Grammar*.

Donald Knuth taught a class at Stanford on *Mathematical Writing* and published a set of lecture notes under the same name (http://bit.ly/LaDKA0).

For help with LaTeX: Oetiker, *The Not So Short Introduction to LaTeX2e* (http://bit.ly/1eFrSPt). David has a very short handout on the most common problems in computer science papers.

Zobel, *Writing for Computer Science.* 2nd edition, 2004. We haven't read this book, but the title sounds really good.

1 General principles

In this section, we will give you three principles that should inform all of your writing and speaking. They seem stultifyingly obvious:

1. Understand your own ideas.
2. Know what a good paper/talk looks like.
3. Write/present for your audience, not yourself.

But each of these principles is really hard to put into practice.

Understand your own ideas.

“If you can’t explain it simply, you don’t understand it well enough.” —attributed to Albert Einstein

If you don’t understand your ideas well, you will not present your ideas well. This is obvious, right? No matter how good of a presenter you are, you cannot get your audience to understand your ideas better than you yourself do.

Conversely, if you don’t present your ideas well, that could be a sign that you don’t understand them well. This means that it can be a good practice to write *in order to* understand. As you struggle to write down your ideas, you will learn to think about them better, and you will find and fix bugs in your ideas. Some people make this a habit, writing a paper about a project from day one.

Know what a good paper/talk looks like.

If you don’t know what a good paper or talk looks like, then you won’t be able to write a good paper or talk. This is also obvious, right? When you see the statements

$$2 + 2 = 4$$
$$2 + 2 = 5$$

how quickly do you see which one is true and which one is false? Instantly? When you see these two images:

how quickly do you see which one is normal and which one is grotesque? Instantly? But when you look at a paper or talk, how quickly can you tell whether it is good or bad? We want you to develop the ability to recognize good and bad communication instantly.

Fortunately, this is not hard to do. Read other people's papers. Not your own papers, because you already (hopefully) understand your own papers. Both published papers and your friends' draft papers. If you understand a paper, it's probably good – ask yourself, what makes this paper good? If you don't understand a paper, it's probably bad – ask yourself, what makes this paper bad?

Go to other people's talks. Both public talks and your friends' practice talks. If you understand a talk, it's probably good – ask yourself, what makes this talk good? If you don't understand a talk, it's probably bad – ask yourself, what makes this talk bad? Of course, don't ask the presenter, "What makes your talk bad?" But do ask the presenter clarification questions. The better you get at this, the better you will be able to ask yourself questions about your own presentations.

Write/present for your audience, not yourself.

It should seem obvious that because the purpose of communication is to impart some knowledge that you have to your audience, your communication should be adapted to the needs of your audience and not yourself. But think about the difference. Writing for yourself is easy: type in a bag of words that activates your neurons just enough to remind you of your own work, done! Writing for others is hard.

Therefore, take the extra time to make your writing/presentation clear to your audience, not just yourself. This time makes all the other time you spend on research not go to waste.

Here are three more, possibly less obvious, principles that will help you write for your audience.

The (n + 1)st revision is better than the nth revision.

"I have rewritten – often several times – every word I have ever written. My pencils outlast their erasers." —Vladimir Nabokov

"Books aren't written – they're rewritten. It is one of the hardest things to accept, especially after the seventh rewrite hasn't quite done it." —Michael Crichton

"I work hard, I work very hard. All the books at least 30 revisions." —Ha Jin

Read your paper/talk over and over again. Analyze each sentence, each section, each slide from the audience's point of view, not your own. Try to predict what the audience will ask, and answer it right there so that they won't have to. Every time you do this, you will spot mistakes in your writing – or, better yet, bugs in your ideas – and the end product will be better.

After you've revised your paper or talk sufficiently, ask your friends to look at it. They will spot dozens of problems that you missed. This may make you feel uncomfortable. But isn't it far better for your friends to see those problems than for the world to see them?

Sometimes backwards is better than forwards.

"A talk is not a murder mystery." —Stuart Shieber

Good flow often puts things in the opposite order from what you might think. You might think you want to say things in logical order or chronological order, but that is not the order that is most useful to the reader. Murder mysteries reveal the identity of the killer at the end. Cookbooks do not reveal the end product at the end. ("Surprise! It's a…foie gras tiramisu!") They show you a picture of the *end* product at the *beginning*. Your papers and talks should be more like cookbooks then murder mysteries.

At the highest level, your paper/talk should state the goal of the work, including the main result or conclusion of the work, and then the rest is about how you reached the goal.

In logical arguments, it's better to first state a claim, then provide the justification for the claim. Mathematical writing is explicitly structured this way (theorems and proofs).

In describing a procedure, first give an example – people extrapolate surprisingly well from one example – then deal with the general case. (But don't skip the general case.)

In explaining a concept, first give the intuition, then give the rigorous explanation. (But don't skip the rigorous explanation.)

Shorter is better than longer.

“I have made this longer than usual because I have not had time to make it shorter.” —Blaise Pascal

In Newfoundland, there is a stretch of highway that is home to many moose and is also very foggy, resulting in many car accidents. In the late 1980s, the authorities erected life-sized, reflective moose-shaped signs to warn drivers of this danger. The moose signs did reduce the number of accidents, but drivers began to slow down to look at the unusual signs or stop to take pictures with them, causing more accidents. The enacted solution was to place new signs a half-mile earlier that read: CAUTION: MOOSE SIGNS AHEAD.[1]

If there is a part of your paper that is unclear, you may be tempted to add an explanation to it to make it more clear. But consider that often the best way to make something less confusing is to delete it.

Homework

Write an elevator pitch that describes who you are and what you are currently working on. Practice reading it aloud. It should be no more than **30 seconds**.

[1] “Moose Signs.” In Robert Finch, *The Iambics of Newfoundland: Notes from an Unknown Shore*, pages 114–116.

2 How to talk about the goal of your work

When you are giving motivation for your work, you should be able to answer these questions, known as the Heilmeier Catechism. They were originally meant for proposals; for papers, the first four questions are more important.

1. What are you trying to do? Articulate your objectives using absolutely no jargon.
2. How is it done today, and what are the limits of current practice?
3. What's new in your approach and why do you think it will be successful?
4. Who cares? If you're successful, what difference will it make?
5. What are the risks and the payoffs?
6. How much will it cost? How long will it take?
7. What are the midterm and final "exams" to check for success?

These are exactly the questions that are in the mind of your intelligent audience. They aren't easy to answer. But if you can't answer them, it's likely you are on the wrong track. Below, we'll refer to these questions as HC1–HC7.

The place to answer these questions is at the beginning of the paper or talk, in the introduction. A typical structure for the introduction is:

- The goal of the work (HC1). Other common elements that will help communicate the goal:
  - An example is very helpful for the reader to understand the goal. If you can continue using this as a *running example* throughout the paper, that's even better.
  - Why is it *good* to achieve the goal (HC4)?
  - Why is it *hard* to achieve the goal (HC2)? This is important for two reasons: to help the reader to understand the problem correctly, and to show the reader that your solution is nontrivial.
  - What is *bad* about the current situation (HC2)? Either what's bad about not having a solution to the problem, or what's bad about current solutions to the problem. This brings up the question of how much to talk about related work.
    - Sometimes necessary for understanding why your work is important. For example: "The standard method for translation of animal sounds is that of MacDonald (1999). However, it only works on text, not audio."
    - Everything else should be relegated to a section at the *end* of the paper. For example: "An approach that is related to ours is that of MacDonald (1999). Their method relies on triangulated dependency propagation, whereas the Gaussian correlations in our method ensure convexity." This doesn't belong here because

the reader doesn't know anything yet about your approach.

- Summary of how you achieve the goal (HC3)
    - You don't need to write, "In Section 2 we describe our method, in Section 3 we describe our experiments, and Section 4 concludes." What's the point of that?
    - It is, however, appropriate to give a summary of the rest of the paper, in terms that anyone in your audience can understand.
    - The key idea that is the basis of your work, stated in an intuitive way
    - Although an element of surprise can be effective, most papers should state their main result(s) in both the abstract and the introduction, and most talks should state their main result(s) within the first few slides. This helps to hold the reader's interest and/or set expectations properly.

The elevator pitch

The so-called elevator pitch is a speech that describes who you are and what you do in 30 seconds. It can be helpful to prepare one so that when you meet someone at a conference, you can tell them quickly why you are an interesting person.

- Introduce yourself by your full name, the same way you write it on your papers.
- HC1 is probably all you have time for, and the no-jargon rule is key.
- Practice it but don't let it sound practiced.

Homework

Imagine that you have invented the Viterbi algorithm. Write the introduction to your paper announcing your discovery. You are free to make up whatever you want about previous approaches.

3 How to talk about your method: concepts

There was a time when it was possible to write a good paper by diving straight from the introduction of the paper into a description of a computer program. For example, Winograd's landmark dissertation on SHRDLU, after a short introduction, proceeds immediately to Section 1.2, "Implementation of the System." But if you were to write such a paper today, you would probably upset the reader; worse, it may be a sign that you have not thought clearly enough about your method.

In some ways, good writing practice is not very different from good programming practice. A well-documented function usually bears some kind of statement of what it is supposed to do, and the function is considered correct only if it actually does what it is supposed to do. Your writing should be the same way: state the *what* before the *how*. This makes your writing more clear, because the reader knows what your method looks like "on the outside" before getting into the internals. It also holds you, the writer, accountable for showing that the *how* is correct.

Normally, you should know what the *what* is before implementing the *how*!

Typical patterns for stating the *what*:

- Pure algorithms paper. A statement of what the algorithm is supposed to do. For example: find the lowest-cost path through a graph. The statement is usually very brief but might be accompanied by explanations of the concepts in it.
- Statistical NLP paper. Nearly always, there is a *model* that fits the goal stated in the introduction, and this leads to objective functions that will be used for *learning* and *inference*. (In Bayesian papers, there is often only inference.)

Bad:

> To solve the natural language generation problem, we use the FOO learning algorithm (Castro, 2006). The FOO algorithm works as follows. We first put all the training examples in a hash table. Then we iterate through the hash table, counting how many hash buckets have multiple entries. Call the proportion of buckets with multiple entries M. If M is prime, then…

Good:

> To solve the natural language generation problem, we take a noisy-channel approach. Imagine that meanings M are generated according to a model $P(M)$, and then transformed into sentences S according to a model $P(S \mid M)$. We express $P(M)$ as a finite-state automaton and $P(S \mid M)$ as a finite-state transducer, shown in Figure 1. Given a corpus of sentences with their meanings, we train the model by searching for transition weights that maximize the likelihood, $P(M)\, P(S \mid M)$.
>
> Finding a global maximum is difficult (Batista, 1997), but the FOO learning

algorithm (Castro, 2006) can quickly find a local maximum. It works as follows…

Modular writing. At a smaller scale, too, it's important to take your complex idea and break it down into "modules" that have a small and easy-to-understand "interface." In the context of describing your ideas, this means defining new terms or notation. For example, the BLEU score is defined as

BLEULoading...

where Loading... is the *n*-gram precision, but, in the face of multiple occurrences of an *n*-gram, becomes messy to define:

> We formalize this intuition as the *modified unigram precision*. To compute this, one first counts the maximum number of times a word occurs in any single reference translation. Next, one clips the total count of each candidate word by its maximum reference count,[2] adds these clipped counts up, and divides by the total (unclipped) number of candidate words.
>
> Modified *n*-gram precision is computed similarly for any *n*: all candidate *n*-gram counts and their corresponding maximum reference counts are collected. The candidate counts are clipped by their corresponding reference maximum value, summed, and divided by the total number of candidate *n*-grams.
>
> > [2] $Count_{clip} = min(Count, Max\_Ref\_Count)$. In other words, one truncates each word's count, if necessary, to not exceed the largest count observed in any single reference for that word.

Here is a more notational version:

> Recall that in information retrieval, if *C* is a set of candidate results and *R* is the set of true results, the *precision* is defined as
>
> Loading....
>
> We generalize this notion to *n*-grams. Let Loading... be the multiset of *n*-grams in string Loading.... (The union of two multisets takes the maximum of their counts, and the intersection takes the minimum.) Then, if *c* is the candidate sentence and *r* is the reference sentence,
>
> Loading...
>
> or if there are multiple references,
>
> Loading...

Homework

What’s wrong with the two paragraphs below? Please write a better paragraph.

(Note for non-NLP people: A language model assigns a probability to any new sequence of English words. We want to assign high probabilities to good English strings, and low probabilities to bad English strings. We typically have a large amount of English sentences to train on.)

1. To solve the language modeling problem, we train a bigram model. We collect, for each word w, a frequency list of words that follow w, then divide each frequency by the sum of all the frequencies. The result is a list of bigram probabilities of the form P(w2 | w1).

(Hint: It’s all “how”. What’s the model? What’s the objective function?)

2. We train a bigram language model as follows. Our data consist of Loading... sentences Loading.... Each sentence Loading... has Loading... words, which we write Loading.... Assume that Loading...<s> and Loading...</s>. Our model is

Loading...

and our goal is to maximize the log-likelihood

Loading...

subject to the constraints

Loading... for all Loading....

The solution is

Loading...,

where Loading... is the Kronecker delta function.

(Hint: It’s almost all “what”. Can you sacrifice some formality in order to increase understandability, eliminate subscripts, eliminate the dreaded Kronecker delta?)

4 How to talk about your method: procedures

We've been telling you for weeks to resist the urge to talk operationally about your method, and now it's time to talk operationally about your method.

There are multiple ways to do this. Some ways are more intuitive and some ways are more precise, and I think a good explanation usually uses more than one, or all of them.

Any of these is appropriate for a paper. In a talk, you want to lean heavily towards the intuitive side of the spectrum.

Pseudocode

Pseudocode is good for stating your method in a very precise way. However, it can never be the sole description of a method, and should never be used in a talk. It should always supplement what you've already said in the text.

Pseudocode shouldn't be too detailed, or else it just looks like code. Python code. It should not take up anywhere near a full page. Ten lines is more digestible.

Neither should pseudocode be too vague; if it is, consider just writing it in plain English.

Specifications

Sometimes, a better alternative to pseudocode is to give a specification that leaves the procedure implicit, yet clear enough to reimplement. For example,

> If there are k events occurring with probabilities $p_1, \ldots, p_k$, the probability that exactly r of them occur is $s(k, r)$, which is defined by the following recurrence:
>
> Loading... if Loading...
> Loading...
> Loading... if Loading...

Although the algorithm isn't stated, this is enough to reimplement the algorithm correctly.

Explanations

A clear explanation in English (and math) is absolutely necessary in a paper; it can't be replaced by pseudocode or an example. In a talk, you can get by with just an example.

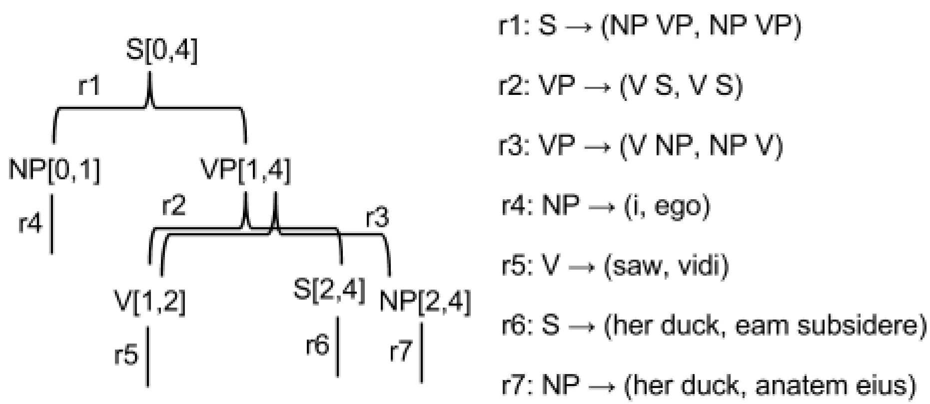
Examples

Examples are vital for helping the reader understand your method. Readers can usually extrapolate easily from the example to the general case. Examples should be consistent throughout the paper/talk as far as possible (e.g., don't switch from Chinese-English translation to English-Chinese).

Pictures

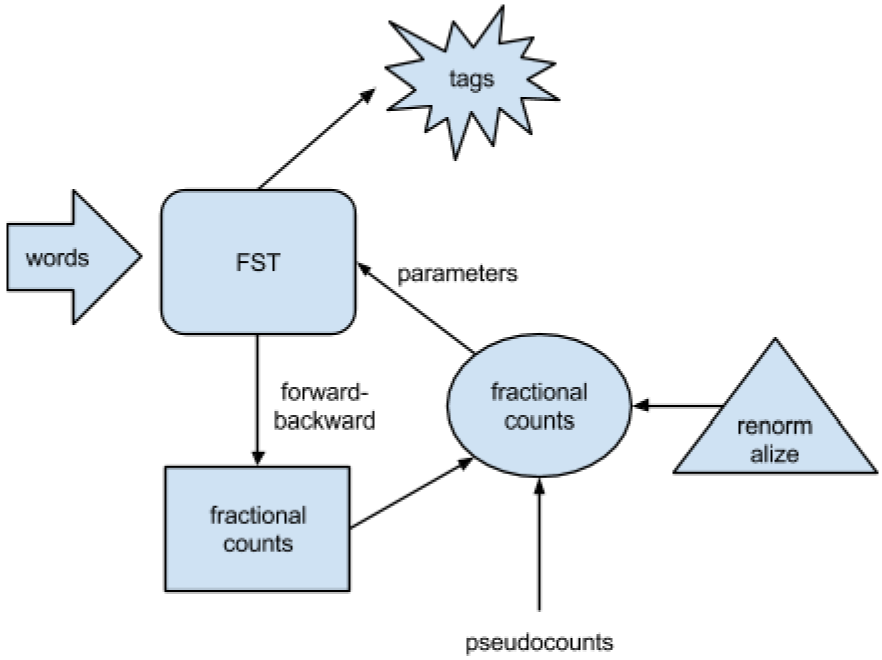
Pictures are one of the most effective ways of getting ideas across, and probably the most difficult to create. In a talk, they're indispensable.

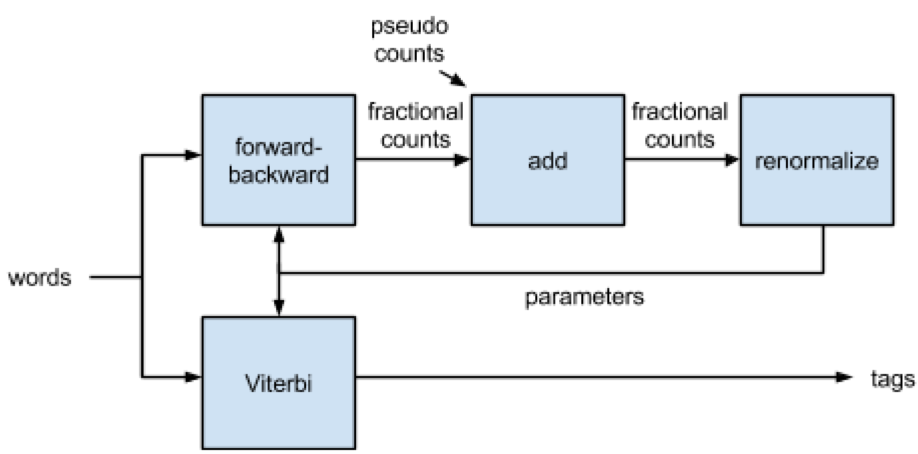
Pictures should make minimal (in a talk: zero) demands on the viewer's memory.



Where does the eye have to travel in order to figure out, say, what the target-language translations are?

Pictures should use a consistent visual "terminology." For example, consider these two illustrations of training an HMM for part-of-speech tagging:





Which figure is better? Which figure do you think took longer to draw?

Homework

Try at least two ways (from the above five) to describe the quicksort algorithm (or something else if you prefer). If you draw a picture of an example, that only counts as one!

5 How to talk about your results

Outline of a typical results section

- Setup. The goal of this subsection should be to describe your work in enough detail that the reader can replicate it. This is a cornerstone of the scientific method. In handouts, it also helps your colleagues to catch bugs.
  - Identify and describe your data in detail. Use examples when helpful.
  - This is *not* the place to describe your method, but it is the place to describe specifics like parameter settings or implementation choices.
  - State the hypothesis that your experiment is supposed to test. Normally, you should in fact know what the hypothesis is before you run the experiment!
- Results. The goal of this subsection is to show that your method is better than a baseline, which is usually the standard or state-of-the-art method for the task. (In the unusual but happy event that your method is the first ever for the task, the baseline should be some naïve method.)
  - Comparisons must always be apples-to-apples, not apples-to-oranges. That is, you should only change one variable at a time.
  - Significance testing is good. But be sure you understand what it means. You can't say "very significant," and you should not mistake "significant" for "large."
  - Be extremely cautious about selectively reporting (cherry-picking) results. It's better to report negative results and explain why they don't hurt your claim.
- Analysis. Here you can put other examples, statistics, or contrastive experiments that will help the reader to understand why, how, or when your method works.

Tables

- A table should always have well-defined rows and columns.
- A table should always have row and column headings.
  - The row heading (which is always on the left) should apply to all the cells to its right. The column heading (which is always on top) should apply to all the cells below it.
  - Headings should be clear, not cryptic (M3+A$^{\text{foo}}$). If you must use abbreviations, you should explain them in the caption.
  - What about the upper-left cell? David's opinion is that it can be either blank or a description of the row headings beneath it – probably not a description of the headings to the right, and never a description of the cells to the lower-right.
- A table should always have interpretable contents.
  - The meaning *and units* of the numbers in a cell should be indicated, usually in the column heading.
  - For large numbers, it seems best to use standard SI suffixes (k=1000, M=1000k, G=1000M, T=1000G). Do not use w=10,000.
  - When describing a data set, it's more useful to report the number of words rather than the number of lines.
  - Don't use excessive precision. For BLEU, report only down to the 0.1% place.

Data summary tables

Whenever you encounter a new data set, build a table to describe it. This table should appear in your first handout.

It's much better to describe data in a table than in text. E.g., consider:

> "We use a portion of the English Gigaword text with 1.1m word tokens and 176k word types, including 101k singletons. The number of sentences is 50,872. We have Spanish translations for 40,325 of these sentences, which include 201k Spanish word types."

Text it makes it hard for readers to instantly compare numbers, e.g., English word types versus Spanish word types. It also obscures the fact that numbers are missing (Spanish word tokens), and other numbers may not be comparable. Putting your data in a table forces you to understand it clearly:

	English	Spanish
# of sentences	40,325	40,325
# of word tokens	0.9m	1.2m
# of word types	159k	201k
# of singleton word types	132k	152k

Don't just put a box around text!

English	1.1m word tokens and 176k word types, including 101k singletons. The numb of sentences is 50,872.
Spanish	40,325 sentences, including 201k Spanish word types.

Empirical results tables

Your goal is to show that your system beats a baseline, so you should almost always have a results table that is some expansion of:

system	**BLEU**
baseline	12.3
ours	23.4

Of course, your table will probably have more rows or columns than this. However, that is never an excuse for breaking the basic pattern. It should be immediately obvious to the reader which numbers are being compared, and to ensure this, all the comparable numbers must line up in a *contiguous* row or column.

system	**setting**	**BLEU**	**METEOR**	**time (s)**
baseline		12.3	32.1	78.9
ours	foo	23.4	43.2	67.8
ours	bar	34.5	54.3	56.7

Comparing along rows instead of columns is less common but also good, for example, if you have many different settings that you want to compare on:

setting	**baseline BLEU**	**our system BLEU**
small data	12.3	23.4
medium data	34.5	45.6
big data	56.7	67.8

Examples of poor tables:

no baseline

system	**BLEU**
ours	23.4

comparable numbers not contiguous

setting	**baseline BLEU**	**baseline seconds**	**our system BLEU**	**our system time**
small data	12.3	78.9	23.4	78.9
big data	34.5	67.8	45.6	67.8

apples to oranges

system	setting	BLEU
baseline	small data	12.3
ours	big data	23.4

But usually the problem arises more subtly...

From recent talk:

slide 31:

System	Setting
ours	big data

slide 33:

System	Accuracy
baseline	57.3
ours	61.2

Astute audience member: “Wait, go back... aren’t you using more data than the baseline system?”

One solution:

System	Setting	Accuracy
baseline	small data	57.3
ours	small data	58.0
ours	big data*	61.2

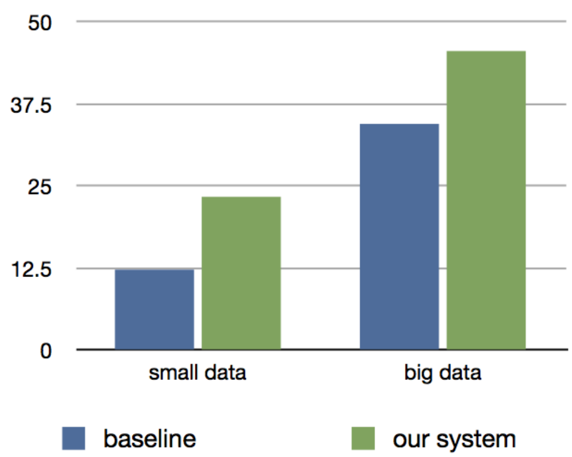
* baseline method is too inefficient to accommodate big data (runtime ~420 CPU years)

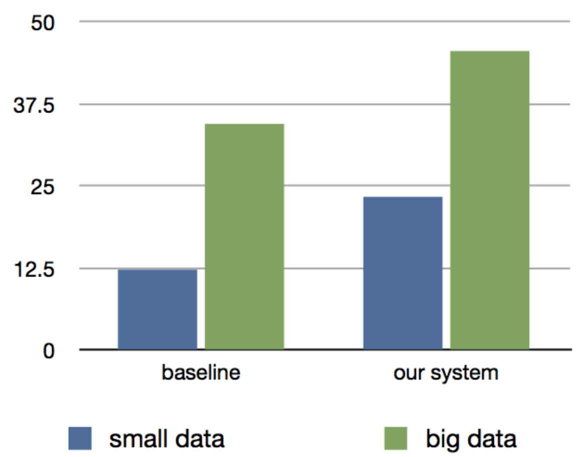
Graphs

- Line graphs
  - Only use a line graph when linear interpolation between data points is meaningful.
  - Kevin's opinion: connect points with straight line segments, not a curvy interpolation
  - Label both axes. Indicate "higher is better" or "lower is better" if not obvious. Avoid a second *y*-axis at all costs.

- Only use 3-D graphs when visualizing 3-D data.

- A bar graph with multiple series should be comparable within each series and within each group. Which of these two makes a more effective comparison?

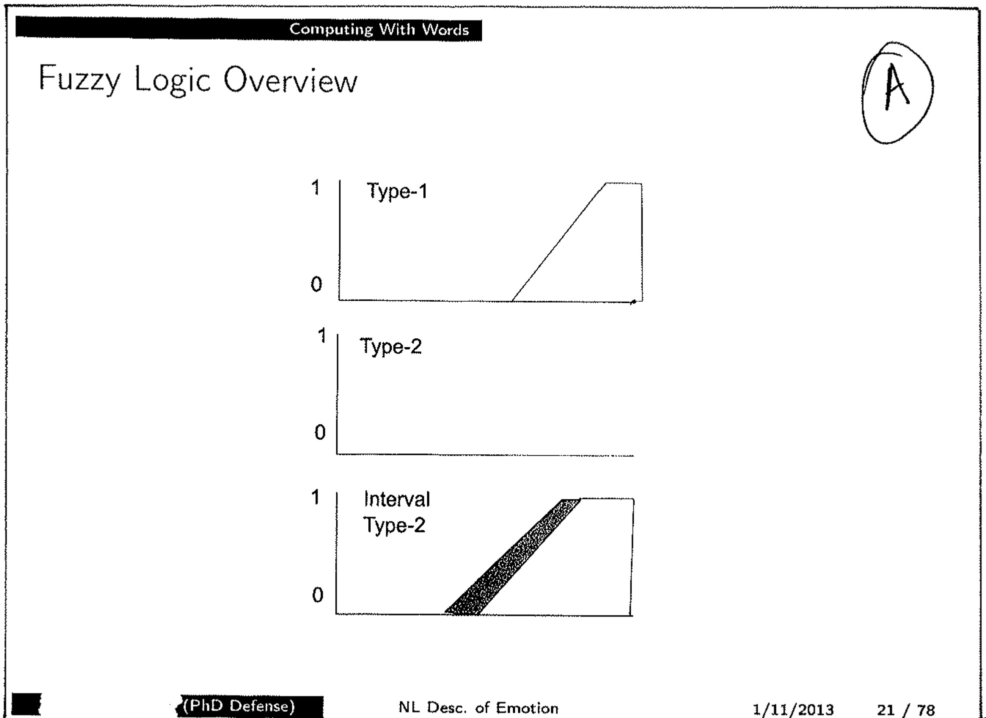




Hint: The first figure says"on small data, we win, and on big data, we also win". The second figure says "for the baseline, big data helps, and for our system, big data helps also".

- To get the best of both worlds of tables and graphs, you can write values on/above the bars.

Discuss these sample tables and graphs. What are strong and weak points?



corpus	size	source
in-domain	100k	zhen
general	1.5M	GALE

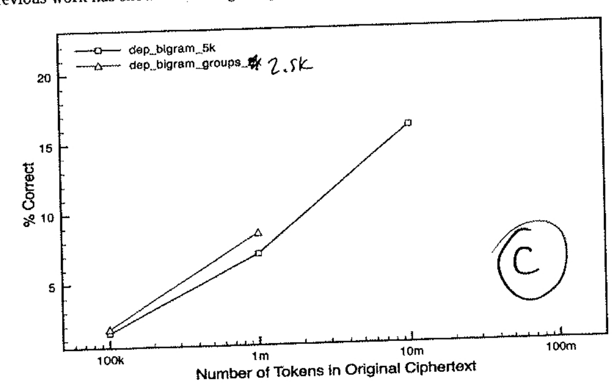
Table 1: LM data set

B

corpus	size	source
train	1.5M	GALE
dev	3k	zhen
test	3k	zhen

Table 2: MT data set

Previous work has shown that using categorized dependency bigrams help decipherment.



PPM-order-4 using new dictionary			Source	target	dict	target'	executable	original	encoded
e: source f: target	Approach 1	original	151476445	174423858	32825282	134406474	229486	325900303	318937687
		compression	30299419	32928924	8992841	26884004		63228343	66405750
		%	20.00%	18.88%	27.40%	20.00%		19.40%	20.82%
	Approach 2	original	151476445	174423858	32825282	139059767	229439	325900303	323590933
		compression	30299419	32928924	8992841	25883859		63228343	65405558
		%	20.00%	18.88%	27.40%	18.61%		19.40%	20.21%
f: source e: target	Approach 1	original	174423858	151476445	28300718	123111735	229486	325900303	326065797
		compression	32928924	30299419	8449491	23122190		63228343	64730091
		%	18.88%	20.00%	29.86%	18.78%		19.40%	19.85%
	Approach 2	original	174423858	151476445	28300718	124582926	229439	325900303	327536941
		compression	32928924	30299419	8449491	22119059		63228343	63726913
		%	18.88%	20.00%	29.86%	17.75%		19.40%	19.46%

D

Homework

1. You aim to develop an improved search method for statistical MT decoding (*). How will you demonstrate success in a single table or graph? Show your proposed table or graph, and list three empirical claims that it justifies in an immediately obvious way.

Hint: first, think hard about what it means to develop a better search method -- what are the benefits you would be seeking, relative to the system currently in use?

> (*) A statistical MT decoder searches through a vast number of target strings to try to find the one with the highest score. This search is necessarily heuristic, as there is no way to exhaustively test all strings.

2. What's wrong with the figure below? Provide a fixed version.

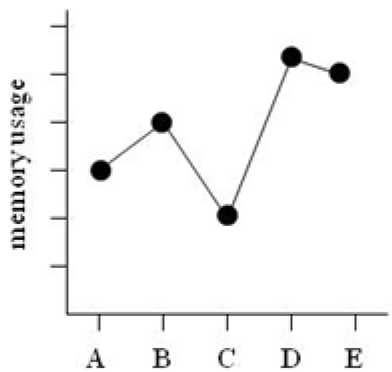


Figure 8. Method C uses the least memory of all methods A-E.

3. Which statement is true?

a) Pie charts should be used whenever you are describing results about food.
b) Scatterplots are best used to show data related to schizophrenia.
c) Plotting software is highly intelligent and knows the best way to present your data.
d) None of the above

4. You get a new bilingual data set, consisting of human translations, which you tokenize (*), align at the sentence level (**), and split into training/development/testing sets. Describe the properties of this data set using a table. You can leave the cells of the table blank.

Hint: you want to describe the corpus in a useful way, while also convincing your audience that you have processed the data correctly.

> (*) Tokenization breaks a string of characters into word tokens. For example,

tokenize(“John’s heart-pump.”) = John ’s heart - pump .

(**) Sentence-level alignment identifies small pieces of text that are translations of each other, for example, one source-language sentence and one target-language sentence. Sometimes humans will translate one source-language sentence as two target-language sentences (or vice-versa), and sometimes they will forget to translate a section of text.

6 How to talk about other people's work

Know other people's work

Before you can talk about other people's work, you must know other people's work. This probably sounds familiar:

- I need to write a related work section and there's only two hours before the deadline
- The reviewer got angry that I didn't cite (Yamazaki, 2000), so I need to stick something in
- I'll just cite all the related work that (Yamazaki, 2012) cites
- I just realized that (Yamazaki, 2006) is really similar, so I need to find some difference

It's much better to know previous work when you *begin* working on a problem, so that you shape your own work accordingly. Then a meaningful interaction with previous work will emerge naturally in your writing.

Bright Line

- What you'll hear: "Is this your idea, or has someone done this before?"
- Novel contributions are the currency of science. You have to draw a bright line between your ideas and others' ideas.

In a talk, an excellent way to draw this line is to show a table like this near the beginning:

	Exact Inference	Polynomial Time	Tested on >10 languages from different families
Yamazaki 2000	no	YES	no
Faiella-Bionen 2004	YES	no	no
Yamazaki & Finch 2008	no	YES	YES
Our work	**YES**	**YES**	**YES**

This makes it clear to the audience exactly what the presenter is contributing.

Nuts and bolts

Related work section

- As advised earlier, an extensive related work section works better near the end of the paper instead of as section 2. Only previous work that is necessary for understanding your work goes in section 1 or 2 (“Background”).

- The purpose of this section is to help readers get a bigger picture of research in your area and how your research fits into it.
- Try to write a story, not a list.

Wrong	Right
The approach of Adams (1999) uses macrobiotic random fields, but her model is generative whereas ours is discriminative. The approach of Baker et al. (2010) uses conditional macrobiotic Boltzmann machines, but their training method is O(n log n) whereas ours is O(n). The approach of Chang (2006) is similar to macrobiotic learning, but does not work on languages other than Turkish.	Most work in this area follows Adams’ original approach (Adams, 1999) using macrobiotic random fields. Others have extended this approach to Boltzmann machines (Baker et al., 2010) and belief propagation (Chang, 2006). A common challenge all these approaches face is that training is computationally expensive. Another line of research, beginning with that of Davidson (2009), uses a mean-field approximation to speed up training. Our method takes this approach as well.

- Or, consider eliminating this section, citing related work right where it is relevant.

Citations

A pet peeve: Composers have “works.” Researchers have “work.”

Wrong	Right
Many previous works have explored the use of macrobiotic machine learning for part-of-speech tagging.	Much previous work has explored the use of macrobiotic machine learning for part-of-speech tagging.

Don’t use citations as noun phrases. You should be able to delete everything in parentheses (or square brackets) and still end up with a grammatical sentence. Although this rule is sometimes annoying, overall it means that you should primarily write about researchers and the research they do, not the papers they write, and this will improve your writing.

Wrong	Right
(Yamazaki, 2004) showed that macrobiotic machine learning is very effective.	Yamazaki (2004) showed that macrobiotic machine learning is very effective.
Macrobiotic machine learning has been used for many NLP tasks, for example, (Yamazaki, 2004) and (Zaretsky, 2006).	Macrobiotic machine learning has been used for many NLP tasks (Yamazaki, 2004; Zaretsky, 2006).
A proof of convergence of macrobiotic EM can be found in (Yamazaki, 2001).	The convergence of macrobiotic EM was proven by Yamazaki (2001).

Yamazaki (2004)'s macrobiotic machine learning framework has been highly influential.	Yamazaki's macrobiotic machine learning framework (Yamazaki, 2004) has been highly influential.
For this task, we used a macrobiotic Markov model (MMM) (Yamazaki, 2004).	For this task, we used a macrobiotic Markov model, or MMM (Yamazaki, 2004).

Footnotes

Footnotes were once primarily used for citations of the sources of the claims in the main text (in other words, metadata, not data). Since we usually use author-date citations now, we don't use footnotes as much. But there are still occasions for metadata[2] that don't fit in a citation.[3]

It's common to bend the rules and put information into a footnote that only very special readers will care about.[4] But never put into a footnote:

- information that is important for understanding the main text[5]
- information that contradicts the main text[6]

[2] Thanks to Jun'ichi Yamazaki for pointing out the connection with macrobiotic machine learning.
[3] LDC catalog numbers LDC2003E14, LDC2004T08, and LDC2005T06.
[4] If X is a manifold, then this is equivalent to the stress-energy tensor in a Riemannian geometry.
[5] Where Q is the number of quantifiers in the training corpus.
[6] Our system used twice as much data as the baseline system.

Papers

The formal structure of a paper forces you to present the big picture of your work in several stages: first, a title (~10 words), then an abstract (~100 words), then, invariably, an introduction section (~1000 words or 1 page).

The title

The title should try to answer question HC1 as succinctly as possible, and using "no jargon" – in research papers or talks (as opposed to proposals), I'd say that this means, "using only terms that are well-known to likely readers." Compare these two titles:

Name-aware Machine Translation
Bilingual Lexical Cohesion Trigger Model for Document-Level Machine Translation

There are a few things that could be done to improve the second title. First, "Trigger" is not a standard technical term in machine translation, and should either be explained in the title or omitted. Less importantly, "Model" is a singular count noun and requires a determiner, and four modifiers ("Bilingual Lexical Cohesion Trigger") is kind of a lot.

The abstract

The abstract is basically a highly compressed version of the introduction, so we don't have much to say about it that isn't in the next section.

But note that the abstract needs to be separable from the rest of the paper, so it should not have any citations or footnotes. If you define an abbreviation in the abstract, you must define it again in the body of the paper.

Talks

What is a talk for?

- Aravind Joshi: a paper is just a written record of your talk
- Mitch Marcus: a talk is just an advertisement to get people to read your paper

Content

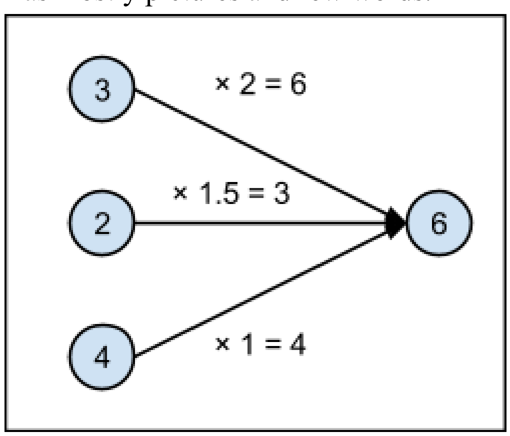
- The main idea is more important than the details
- Impact is more important than comprehensiveness
- Intuition is more important than rigor
- Examples are better than generalizations

What is a talk?

- You, talking

Slides

- Exist only to communicate what you cannot communicate through words.
    - They are not your notes.
    - It’s true that slides also serve as a transcript to share with people later.
- Therefore a good slide has mostly pictures and few words.



- Viterbi algorithm
    - dynamic programming
- Each edge's Viterbi weight is tail node's Viterbi weight times edge weight
- Each node's Viterbi weight is max of incoming edge's Viterbi weight

Preparation

- Learn how to use your presentation software, in particular, how to jump to an arbitrary slide without advancing through all your slides, and especially without repeating all your animations.
- Many people use some kind of remote control.
- If your computer requires a VGA adapter, keep it with your computer at all times.
- You don't need a laser pointer. Shining red lights onto words makes them way harder to read. Plus, 99% of the time, it's obvious what you're referring to. Audiences hate laser pointers. Try this experiment: at the beginning of your talk, pick up a laser pointer and say, “Hmm, it seems to be broken.” Then look at your audience, and you'll see how happy they are! Exception: Use a laser pointer when you practice at home with your cat. Another exception: Use it when you practice with James Bond. When James Bond says, “You expect me to read your slides with that crazy red light bouncing all around?” you say, “No, Mr. Bond, I expect you to die!!” and you point the laser right into his eyes.

Practice

- Do it

Delivery

- Don’t read your slides.
    - Exception: If you put up a quotation or cartoon in a slide presentation, read the text out loud, in full, without summarizing. Three reasons: (1) You will inevitably start talking before the audience has finished reading, and they can't process both things; (2) Someone will start laughing within three seconds, making others feel bad they are still reading; (3) By keeping everyone together, you focus your impact..
- Enthusiasm is contagious.
- It is much better, and much rarer, to finish early than late.

Homework

Present a 3-minute talk fragment (one slide, possibly with builds or animations) explaining mergesort.

Posters

A good talk probably also makes a good poster. Exception: unlike slides, a poster should probably be designed to be intelligible without you standing in front of it. But a great talk probably doesn't make a great poster. Why?

Your audience is usually a single person. So don't give every single person the same speech. Based on his/her level of knowledge (just ask if you don't know), tailor your presentation: short or long, elementary or advanced, focusing on one part of the work or another, etc. You can prepare several versions of your speech, but be prepared to improvise too.

Your visitor might:

- Want to hear your standard speech
- Just look at your title and moves on
- Read through your whole poster and then asks questions
- Have a long conversation with you
- Listen to your conversation with someone else

A poster is a big piece of paper. Compared with slides, there's one disadvantage: you can't use animations (but I suppose you could use an iPad to show an animation if it's important).

On the other hand, talk slides are limiting. They're all the same size and each contains only a little bit of information, and you can only look at one at a time and in a fixed sequence. It's not easy to make clear the structure of your work: all its parts and how they relate to each other. But on a poster, you don't have any of these limitations.

So don't just put a bunch of slides on your poster; take advantage of the format. For example, make your big idea big, in the center of the poster, and make all the supporting ideas (motivation, background, detailed method, results, related work, etc.) surround it.

A final suggestion: print out (color) copies of your poster and give them out.

Spectral Learning of Deep Neural Networks for Lagrangian Relaxation

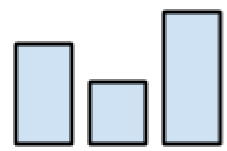
- Lorem ipsum dolor sit amet
- Consectetuer adipiscing elit
- Sed diam nonummy nibh eiusmod

- Lorem ipsum dolor sit amet
- Consectetuer adipiscing elit
- Sed diam nonummy nibh eiusmod

- Lorem ipsum dolor sit amet
- Consectetuer adipiscing elit
- Sed diam nonummy nibh eiusmod

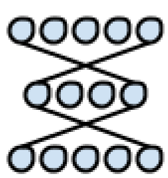
- Lorem ipsum dolor sit amet
- Consectetuer adipiscing elit
- Sed diam nonummy nibh eiusmod

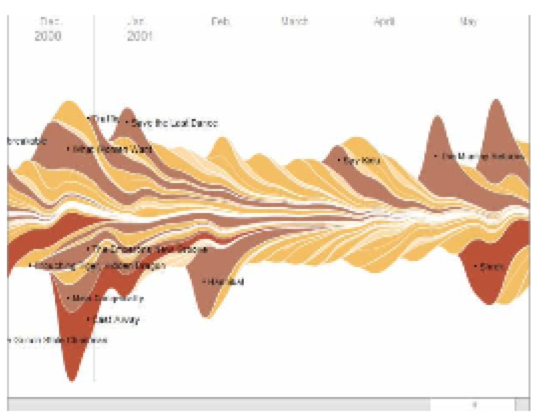
- Lorem ipsum dolor sit amet
- Consectetuer adipiscing elit
- Sed diam nonummy nibh eiusmod

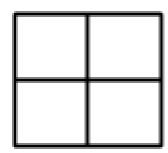


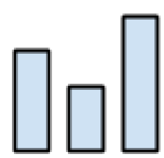
Spectral Learning of Deep Neural Networks for Lagrangian Relaxation

- Lorem ipsum dolor sit amet
- Consectetuer adipiscing elit
- Sed diam nonummy nibh eiusmod









Proposals

Do you have some cool research you want to do?

1) Propose it
2) People say OK, or give you money
3) Do it

The PhD proposal is usually the only chance you have to practice this important research skill. It is a kind of contract you negotiate with your committee. If you do what's in the proposal document, you will get a PhD! It's a one-way contract, in your favor -- if you encounter interesting things in the course of your research, you are free to change direction.

The Heilmeier Catechism. If you can answer these questions, you are set!

1. What are you trying to do? Articulate your objectives using absolutely no jargon.
2. How is it done today, and what are the limits of current practice?
3. What's new in your approach and why do you think it will be successful?
4. Who cares? If you're successful, what difference will it make?
5. What are the risks and the payoffs?
6. How much will it cost? How long will it take?
7. What are the midterm and final "exams" to check for success?

Proposal structure. There are many ways to structure a proposal.
Here's one:

- We'd like to be able to do <X = some amazing thing, like establish a Mars colony>.
- Doing X would require doing <A = really hard thing>, <B = hard thing>, and <C = hard thing>
- Luckily, Yamamoto and Fitz (2011) just solved B!
- I can solve C, which puts us closer to X. So this proposal is about C.
- (A is still future work.)
- Why haven't people solved C already?
    - The first person to try C was Woody (1967).
    - Current state-of-the-art is Blick and Aaron (2009)
    - C is hard because of challenge W.
- etc.

Here's another:

- Low-density languages are hard for statistical MT
- Prior work includes ...
- I invent three novel techniques, and prove that they significantly increase Bleu score in low-density MT:
    - I improve word alignment by …
    - I exploit non-parallel corpora by …
    - My novel vowel-removal procedure ...

List of proposed scientific contributions

This is the most often-overlooked component of a proposal. It is a short bullet list, usually at the end of the proposal's introduction. It answers the question: "How will science be different after you have successfully carried out the proposed work?"

Avoid vague statements like
- "I will work on area X."
- "I will build an accurate parser."

Better:
- "I improve the accuracy of Kinyarwanda-English machine translation by 6.2 Bleu points over previous results."
- "I integrate phrase-based, syntax-based, and semantics-based machine translation into a single, novel mathematical framework. Exploiting this framework, I develop a generic decoding algorithm that runs 5 times faster than previous methods.

3-4 bullet items is good, more is too diffuse. It's hard work, but if your contributions are crisp and clear, your proposal will often sail through. On the other hand, if readers flip through the proposal and can't find these bullets, then you are in trouble.

Outside the scope

It's helpful to tell the reader what you are not going to do. This gives your committee comfort that you are not proposing to solve all the world's problems.

Appendix: The Marina Lexicon

A famous science fiction writers' workshop developed a list of common fiction-writing errors, called "The Turkey City Lexicon" (http://bit.ly/1j9qWsn).

Beginning writers make all the errors on that list. Having a name for each error makes it easy to offer constructive criticism like this:

> Reader: "At the top of page 3, you have 'Squid in the Mouth'."
> Author: "You're right!"

Here's a similar list for scientific communication.

Main Point Missing

- What you'll hear: "OK, now I get it … your main point is X. But, X doesn't actually appear anywhere on the page."
- After a long in-person discussion with the author, the reader finally figures out what the main point of Section 2 is. The author is then shocked to realize that the main point does not appear in print anywhere -- it was only in his/her mind.

Murder Mystery

- What you'll see: People looking at their smartphones.
- Reason: The main point is there, but is kept secret until the end, leaving the audience with no sense of where it is going.

42

- What you'll hear: "That's a great solution! But what's the problem?"
- The author has presented a technically impeccable, well-described solution. But the audience doesn't know what problem is being solved. Either the author forgot to make that clear, or there is no significant problem. Scientists are much better at finding good solutions than finding good problems.
- Authors sometimes try this:
    - Slide 1. I'm going to introduce "Information Density Reconstruction".
    - Slide 2. The first question is, why do we need "Information Density Reconstruction"?

  This approach seems natural to the speaker, who can't resist highlighting the new solution. But the audience needs it the other way around. Start with an important, challenging problem, then present previous work on it, and then present the weaknesses of that previous work… and *then* say "Here's the solution, which I call Information Density Reconstruction!" At that point, the audience is fired up to learn more.

Motivation Problem

- What you'll hear: "I don't understand what you are trying to do."
- Scientists are usually deep in their own weeds. When a science project begins, scientists know what they are doing, and why. Then they start focusing on the details. Pondering the big picture is no longer part of their daily routine. So "what" and "why" get mentally pushed down further and further, replaced by "how". So when scientists have to explain what they are doing, they naturally start explaining *what they are doing that day*, because it's hard to access information that has been pushed so far down.

What Before How

- What you'll hear: "What are the inputs, and what are the outputs?"
- The author has launched into a full-blown description of some process or algorithm, but the audience doesn't know what its high-level behavior is supposed to be. Give sample input/output pairs before going into the process itself.

Bright Line

- What you'll hear: "Is this your idea, or has someone done this before?"
- Novel contributions are the currency of science. You have to draw a bright line between your ideas and others' ideas. I recently sat through a talk on crowdsourcing, by someone who works in crowdsourcing. At the end of the talk, I was impressed -- it seemed like that person had done a lot! But then I realized that there are tons of people working on crowdsourcing, so most (all?) of that work had to be other people's.

Ramble On

- What you'll hear: "I'm completely lost. You were talking about X, then you moved to Y, and now Z … what's the connection? X, Y, and Z seem like completely different things."
- A common problem. The solution depends on the circumstance. You may need to cut superfluous digressions, so that the logical flow is more apparent. You may need to provide a roadmap at the beginning, and refer back to that roadmap as you go.
- In the best case, you may actually have three good (but disconnected) ideas. You can develop these into three separate papers.
- In the worst case, you are rambling from topic to topic because you don't actually have a main idea.

Moose Signs Ahead

- An explanation added to something unclear that makes it even more unclear
- Solution: delete or rewrite what was unclear in the first place.

Obviously False

- What you'll hear: "This sentence seems obviously false. You must mean something else."
- In geometry class, you may have learned that "A square is a rectangle." This is obviously false to a normal person, which is why the geometry teacher needs to supply lots of definitions beforehand. Obviously false statements appear surprisingly often in scientific communication,

usually without the necessary definitions (or time for the audience to process them). It's best to avoid them.

Inconsistent Numbers

- What you'll hear: "These two numbers aren't consistent with each other."
- Suppose you are describing some text corpus. You say that the corpus is 435 words long, with 615 distinct words. Those two numbers aren't consistent. More often, it takes significant inference to determine that two numbers aren't consistent with each other. The analysis then reveals a logic bug or a wrongly-designed experiment. Result: the audience no longer believes any of the numbers in the presentation.

Variable Undefined

- What you'll hear: "This variable was never defined."
- The author writes X to refer the number of word types in the English half of the bilingual corpus, but then forgets to tell the audience what X stands for. Write definitions before formulas. Sometimes an author will give a long formula involving T, and only afterwards say "where T is the running time". Readers will try hard to puzzle out the meaning of the formula and finally fail, just before the author tells says: "Ha! Here's the missing piece."

Is That the Same X?

- What you'll hear: "You used the same variable twice to mean two different things."
- Always an inadvertent mistake. Revision will help you catch these.

Humpty Dumpty

- What you'll hear: "You use the term X. But X already has a different established technical meaning."
- Don't use the term "antecedent" to mean "any word before this one". That term already has a specific meaning, and the audience can't recover from the wrong usage. Don't be like Humpty Dumpty, who says "When I use a word, it means just what I choose it to mean!"

Axis of Evil

- What you'll hear: "What is the *x* axis, and what is the *y* axis?"
- Everything in a graph needs to be labeled. Axes should have adequate text descriptions plus units -- for example, "Decoding time (seconds)" rather than "Time". Figures should be self-standing. There should be enough information in the caption and the graphic for a reader to understand the data without reference to the text.
- Tables do not have axes, they have rows and columns. So we don't refer to column headers as "the x-axis of the table".
- An axis representing percentage accuracy cannot have tick marks going past 100. A reader who sees the "120" will draw all sorts of wrong conclusions about the graph. Plotting software is not intelligent!