Natural Language Processing: Machine Translation

Christopher Manning

Borrows some slides from Kevin Knight, Dan Klein, and Bill MacCartney

Lecture Plan

1. The IBM (Alignment) Models [30 mins]
2. Middle 10 mins: Administration, questions, catch up [10 mins]
3. Getting parallel sentences to train on [10 mins]
4. Searching for the best translation: Decoding [10 mins]
5. MT Evaluation [10 mins]

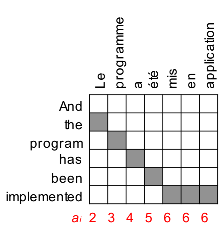
Lecture Plan

1. **The IBM (Alignment) Models** [30 mins]
2. Middle 10 mins: Administration, questions, catch up [10 mins]
3. Getting parallel sentences to train on [10 mins]
4. Searching for the best translation: Decoding [10 mins]
5. MT Evaluation [10 mins]

IBM Models 1,2,3,4,5

- Models for P(**f**|**e**) and P(**a**|**f**,**e**) via P(**f**,**a**|**e**)
- There is a set of English words and the extra English word NULL
- Each English word generates and places 0 or more French words
- Any remaining French words are deemed to have been produced by NULL ("spurious words")
- Some English words may not be used at all ("zero fertility words")

IBM Model 1 parameters



$$P(f,a|e) = P(m|\ell)\prod_i P(a_i)t(f_i|e_{a_i})$$

$$= \epsilon \prod_i P(a_i)t(f_i|e_{a_i})$$

$$= \epsilon \prod_i \frac{1}{\ell+1} t(f_i|e_{a_i})$$

$$= \frac{\epsilon}{(\ell+1)^m} \prod_i t(f_i|e_{a_i})$$

Model 1: Word alignment learning with Expectation-Maximization (EM)

- Start with $t(f^p|e^q)$ uniform, including P(f^p|NULL)
- For each sentence pair (e, f)
  - For each French position i

E

    - Calculate posterior over English positions P(a_i | e, f)

$$P(a_i = j|f,e) = \frac{t(f_i|e_j)}{\sum_{j'} t(f_i|e_{j'})}$$

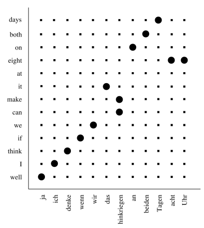
    - Increment count of word f_i translating each word e_{a_i}
      - C(f_i|e_j) += P(a_i = j | f, e)

M

- Renormalize counts to give probs $t(f^p|e^q) = \frac{C(f^p|e^q)}{\sum_{f^x} C(f^x|e^q)}$
- Iterate until convergence

IBM Models 1,2,3,4,5

- In Model 2, the placement of a word in the French depends on where it was in the English



- Unlike Model 1, Model 2 captures the intuition that translations should usually "lie along the diagonal"
- A main focus of PA #1
- See Collins (2011).

Applying Model 1*

P(*f*, *a* | *e*) can be used as a *translation model* or an *alignment model*

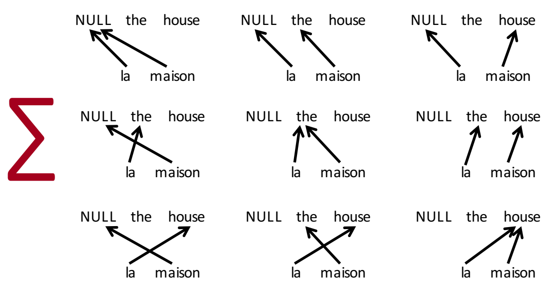
As translation model $P(f|e) = \sum_a P(f,a|e)$

As alignment model $P(a|e,f) = \frac{P(f,a|e)}{P(f|e)}$

$= \frac{P(f,a|e)}{\sum_{a'} P(f,a'|e)}$

* Actually, any *P*(*f*, *a* | *e*), e.g., any IBM model

Summing out alignments



IBM Models 1,2,3,4,5

- In Model 3, we model how many French words an English word can produce, using a concept called *fertility*

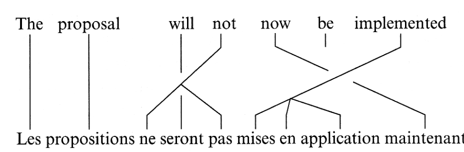
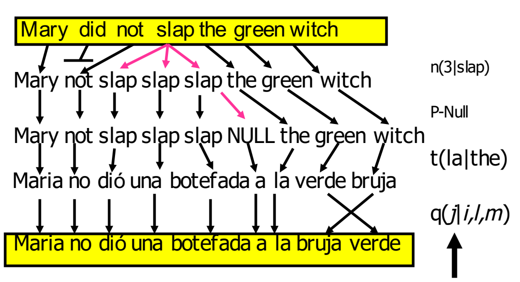


Figure 32.3
Alignment example.

Model 3 generative story



Probabilities can be learned from raw bilingual text.

IBM Model 3 (from Knight 1999)

- For each word e_j in English sentence, choose a fertility ϕ_j. The choice of ϕ_j depends only on e_j, not other words or ϕ's: $n(\phi_j \mid e_j)$
- For each word e_j, generate ϕ_j French words. Choice of French word depends only on English word e_j, not on English context or any other French words.
- Permute all the French words. Each French word gets assigned absolute target position slot (1,2,3, etc.). Choice of French word position dependent only on absolute position of English word generating it and sentence lengths

Model 3: P(f|e) parameters

- What are the parameters for this model?
- Word translation: t(casa | house)
- Spurious words: t(f_i | NULL)
- Fertilities: n(1|house): prob that "house" will produce 1 Spanish word whenever it appears.
- Distortions: q(5|2,4,6): prob that word in position 2 of French translation was generated by word in position 5 of English sentence, given that 4 is length of English sentence, 6 is French length

Spurious words

- We could have n(3|NULL) (probability of there being exactly 3 spurious words in a French translation)
  - But seems wrong…
- Instead, of n(0|NULL), n(1|NULL) … n(25|NULL), have a single parameter p_1
- After assign fertilities to non-NULL English words we want to generate (say) z French words.
- As we generate each of z words, we optionally toss in spurious French word with probability p_1
- Probability of not adding spurious word: $p_0 = 1 - p_1$

Distortion probabilities for spurious words

- Shouldn't just have q(0|5,4,6), i.e., chance that source position for word 5 is position 0 (NULL).
- Why? These are spurious words! Could occur anywhere!! Too hard to predict
- Instead,
  - Use normal-word distortion parameters to choose positions for normally-generated French words
  - Put NULL-generated words into empty slots left over
  - If three NULL-generated words, and three empty slots, then there are 3!, or six, ways for slotting them all in
  - We'll assign a probability of 1/6 for each way!

Model 3 parameters

- n, t, p, q
- Again, if we had complete data of English strings and step-by-step rewritings into Spanish, we could:
  - Compute n(0|did) by locating every instance of "did", and seeing how many words it translates to
  - t(maison|house) how many of all French words generated by "house" were "maison"
  - q(5|2,4,6) out of all times some second word is in a translation, how many times did it come from the fifth word (in sentences of length 4 and 6 respectively)?

Since we don't have word-aligned data…

- We bootstrap alignments from incomplete data
- From a sentence-aligned bilingual corpus
  1) Assume some startup values for n, q, t, p.
  2) Use values for n, q, t, p in model 3 to work out chances of different possible alignments. Use these alignments to update values of n, q, t, p.
  3) Go to 2
- This is a more complicated case of the EM algorithm

Difficulty: Alignments are no longer independent of each other. Have to use approximate inference

Examples: translation & fertility

the

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
le	0.497	1	0.746
la	0.207	0	0.254
les	0.155		
l'	0.086		
ce	0.018		
cette	0.011		

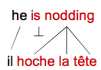
not

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
ne	0.497	2	0.735
pas	0.442	0	0.154
non	0.029	1	0.107
rien	0.011		

farmers

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
agriculteurs	0.442	2	0.731
les	0.418	1	0.228
cultivateurs	0.046	0	0.039
producteurs	0.021		

Example: idioms



nodding

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
signe	0.164	4	0.342
la	0.123	3	0.293
tête	0.097	2	0.167
oui	0.086	1	0.163
fait	0.073	0	0.023
que	0.073		
hoche	0.054		
hocher	0.048		
faire	0.030		
me	0.024		
approuve	0.019		
qui	0.019		
un	0.012		
faites	0.011		

Example: morphology

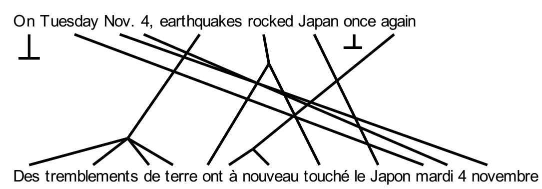
should

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
devrait	0.330	1	0.649
devraient	0.123	0	0.336
devrions	0.109	2	0.014
faudrait	0.073		
faut	0.058		
doit	0.058		
aurait	0.041		
doivent	0.024		
devons	0.017		
devrais	0.013		

IBM Models 1,2,3,4,5

- In model 4 the placement of later French words produced by an English word depends on what happened to earlier French words generated by that same English word

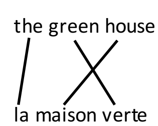
Alignments: linguistics



IBM Models 1,2,3,4,5

- In model 5 they patch model 4. They make it do non-deficient alignment. That is, you can't put probability mass on impossible things.

Alignments: linguistics



- There isn't enough linguistics to explain this pattern within the translation model
- Have to depend on the language model to get it right
- That may be unrealistic
- And may be harming our translation model … and final system

IBM StatMT Translation Models

- IBM1 – lexical probabilities only
- IBM2 – lexicon plus absolute position
- HMM – lexicon plus relative position
- IBM3 – plus fertilities
- IBM4 – inverted relative position alignment
- IBM5 – non-deficient version of model 4

- All these models handle 0:1, 1:0, 1:1, 1:n alignments *only*

[Brown et al. 93, Vogel et al. 96]

Why all the models?

- We don't start with aligned text, so we have to get initial alignments from somewhere.
- The alignment space has many local maxima
- Model 1 is words only, a simple model that is relatively easy and fast to train.
- The output of M1 can be a good place to start M2
  - "Starting small". Also, it's convex!
- The sequence of models allows a better model to be found, faster
  - The intuition is like deterministic annealing … or the pre-training done in Deep Learning

Lecture Plan

1. The IBM (Alignment) Models [30 mins]
2. **Middle 10 mins: Administration, questions, catch up** [10 mins]
3. Getting parallel sentences to train on [10 mins]
4. Searching for the best translation: Decoding [10 mins]
5. MT Evaluation [10 mins]

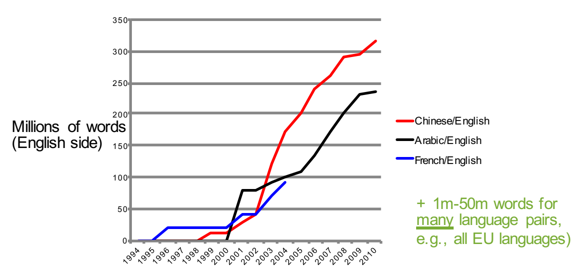
Lecture Plan

1. The IBM (Alignment) Models [30 mins]
2. Middle 10 mins: Administration, questions, catch up [10 mins]
3. **Getting parallel sentences to train on** [10 mins]
4. Searching for the best translation: Decoding [10 mins]
5. MT Evaluation [10 mins]

Getting Parallel Sentence Data

- Hard way:
  - Create your own data
  - Find and collect parallel data from web
- Easy way: Use existing curated data
  - Linguistic Data Consortium (LDC)
    - http://www.ldc.upenn.edu/
  - EuroParl/WMT:
    - http://www.statmt.org/europarl/
    - Around 50 million words per language for "old" EU countries

Ready-to-Use Online Bilingual Data



(Data stripped of formatting, in sentence-pair format, available from the Linguistic Data Consortium at UPenn).

Tokenization (or Segmentation)

- English
  - Input (some character stream):

    `"There," said Bob.`
  - Output (7 “tokens” or “words”):

    `" There , " said Bob .`
- Chinese
  - Input (char stream): 美国关岛国际机场及其办公室均接获一名自称沙地阿拉伯富商拉登等发出的电子邮件。
  - Output: 美国 关岛国 际机 场 及其 办公室均接获 一名 自称 沙地 阿拉 伯富 商拉登 等发 出 的 电子邮件。

Sentence Alignment

The old man is happy. He has fished many times. His wife talks to him. The fish are jumping. The sharks await.

El viejo está feliz porque ha pescado muchas veces. Su mujer habla con él. Los tiburones esperan.

Sentence Alignment

1. The old man is happy.
2. He has fished many times.
3. His wife talks to him.
4. The fish are jumping.
5. The sharks await.

1. El viejo está feliz porque ha pescado muchas veces.
2. Su mujer habla con él.
3. Los tiburones esperan.

Sentence Alignment

1. The old man is happy.
2. He has fished many times.
3. His wife talks to him.
4. The fish are jumping.
5. The sharks await.

1. El viejo está feliz porque ha pescado muchas veces.
2. Su mujer habla con él.
3. Los tiburones esperan.

Done by similar Dynamic Programming or EM: see FSNLP ch. 13 for details

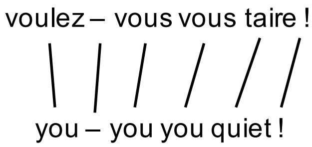
Lecture Plan

1. The IBM (Alignment) Models [30 mins]
2. The Middle 10: Course administration, random questions, catch up, or get a head start on the back 30 [10 mins]
3. Getting parallel sentences to train on [10 mins]
4. **Searching for the best translation: Decoding** [10 mins]
5. MT Evaluation [10 mins]

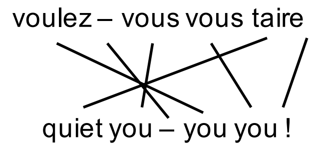
Search for Best Translation

voulez – vous vous taire !

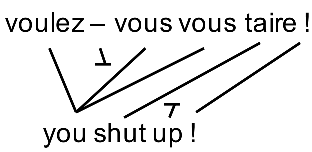
Search for Best Translation



Search for Best Translation



Search for Best Translation



Searching for a translation

Of all conceivable English word strings, we want the one maximizing P(e) x P(f | e)

Exact search

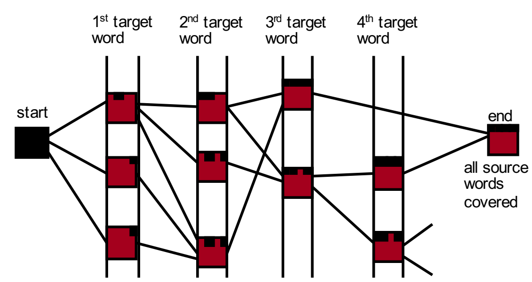
- Even if we have the right words for a translation, there are **n!** permutations.
- We want the translation that gets the highest score under our model
- Finding the argmax with a n-gram language model is NP-complete [Germann et al. 2001].
- Equivalent to Traveling Salesman Problem

40

Searching for a translation

- Several search strategies are available
  - Usually a beam search where we keep multiple stacks for candidates covering the same number of source words
  - Or, we could try “greedy decoding”, where we start by giving each word its most likely translation and then attempt a “repair” strategy of improving the translation by applying search operators (Germann et al. 2001)
- Each potential English output is called a *hypothesis*.

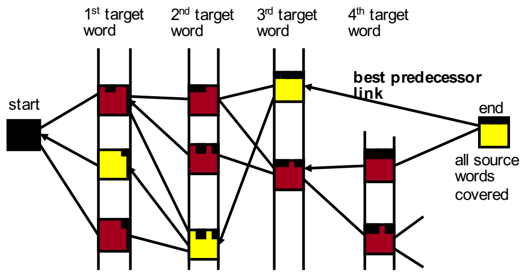
Dynamic Programming Beam Search



Each partial translation hypothesis contains:
- Last English word chosen + source words covered by it
- Next-to-last English word chosen
- Entire coverage vector (so far) of source sentence
- Language model and translation model scores (so far)

[Jelinek, 1969; Brown et al, 1996 US Patent; (Och, Ueffing, and Ney, 2001]

Dynamic Programming Beam Search



Each partial translation hypothesis contains:
- Last English word chosen + source words covered by it
- Next-to-last English word chosen
- Entire coverage vector (so far) of source sentenc
- Language model and translation model scores (so far)

[Jelinek, 1969; Brown et al, 1996 US Patent; (Och, Ueffing, and Ney, 2001]

Lecture Plan

1. The IBM (Alignment) Models [30 mins]
2. Middle 10 mins: Administration, questions, catch up [10 mins]
3. Getting parallel sentences to train on [10 mins]
4. Searching for the best translation: Decoding [10 mins]
5. **MT Evaluation** [10 mins]

Evaluating Alignments: Alignment Error Rate (Och & Ney 2000)

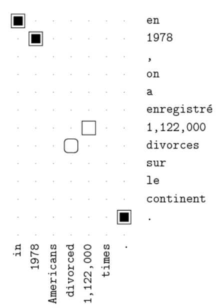
□ = Sure

○ = Possible

■ = Alignments (predicted)

$$AER(A,S,P) = \left(1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}\right)$$

$$= \left(1 - \frac{3+3}{3+4}\right) = \frac{1}{7}$$



Most work has used AER and we do, but it is problematic, and it's better to use an alignment F measure (Fraser and Marcu 2007)

Comparative results (AER)

[Och & Ney 2003]

Model	Training scheme	Size of training corpus 0.5K	8K	128K	1.47M
Dice		50.9	43.4	39.6	38.9
Dice+C		46.3	37.6	35.0	34.0
Model 1	1^5	40.6	33.6	28.6	25.9
Model 2	$1^5 2^5$	46.7	29.3	22.0	19.5
HMM	$1^5 H^5$	26.3	23.3	15.0	10.8
Model 3	$1^5 2^5 3^3$	43.6	27.5	20.5	18.0
	$1^5 H^5 3^3$	27.5	22.5	16.6	13.2
Model 4	$1^5 2^5 3^3 4^3$	41.7	25.1	17.3	14.1
	$1^5 H^5 3^3 4^3$	26.1	20.2	13.1	9.4
	$1^5 H^5 4^3$	26.3	21.8	13.3	9.3
Model 5	$1^5 H^5 4^3 5^3$	26.5	21.5	13.7	9.6
	$1^5 H^5 3^3 4^3 5^3$	26.5	20.4	13.4	9.4
Model 6	$1^5 H^5 4^3 6^3$	26.0	21.6	12.8	8.8
	$1^5 H^5 3^3 4^3 6^3$	25.9	20.3	12.5	8.7

Common software: GIZA++/Berkeley Aligner

Illustrative translation results

- *la politique de la haine .* *(Foreign Original)*
- politics of hate . (Reference Translation)
- the policy of the hatred . (IBM4+N-grams+Stack)

- *nous avons signé le protocole .* *(Foreign Original)*
- we did sign the memorandum of agreement . (Reference Translation)
- we have signed the protocol . (IBM4+N-grams+Stack)

- *où était le plan solide ?* *(Foreign Original)*
- but where was the solid plan ? (Reference Translation)
- where was the economic base ? (IBM4+N-grams+Stack)

对外经济贸易合作部今天提供的数据表明，今年至十一月中国实际利用外资四百六十九点五九亿美元，其中包括外商直接投资四百点零七亿美元。

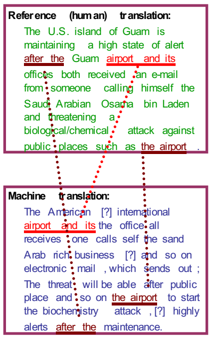
the Ministry of Foreign Trade and Economic Cooperation, including foreign direct investment 40.007 billion US dollars today provide data include that year to November china actually using foreign 46.959 billion US dollars and

MT Evaluation

- Manual (the best!?):
  - SSER (subjective sentence error rate)
  - Correct/Incorrect
  - **Adequacy and Fluency** (5 or 7 point scales)
  - Error categorization
  - **Comparative ranking of translations**
- Testing in an application that uses MT as one sub-component
  - E.g., question answering from foreign language documents
    - May not test many aspects of the translation (e.g., cross-lingual IR)
- Automatic metric:
  - WER (word error rate) – why problematic?
  - **BLEU (Bilingual Evaluation Understudy)**

BLEU Evaluation Metric

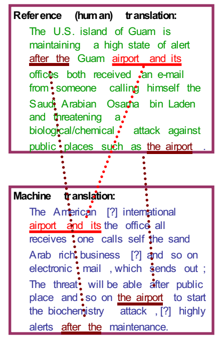
(Papineni et al, ACL-2002)



- N-gram precision (score is between 0 & 1)
  - What percentage of machine n-grams can be found in the reference translation?
    - An n-gram is an sequence of n words
  - Not allowed to match same portion of reference translation twice at a certain n-gram level (two MT words *airport* are only correct if two reference words *airport*; can't cheat by typing out "the the the the the")
  - Do count unigrams also in a bigram for unigram precision, etc.
- Brevity Penalty
  - Can't just type out single word "the" (precision 1.0!)
- It was thought quite hard to "game" the system (i.e., to find a way to change machine output so that BLEU goes up, but quality doesn't)

BLEU Evaluation Metric

(Papineni et al, ACL-2002)



- BLEU is a weighted geometric mean, with a brevity penalty factor added.
  - Note that it's precision-oriented
- BLEU4 formula

  (counts n-grams up to length 4)

exp (1.0 * log p1 +
0.5 * log p2 +
0.25 * log p3 +
0.125 * log p4 –
max(words-in-reference/words-in-machine – 1, 0)

p1 = 1-gram precision
P2 = 2-gram precision
P3 = 3-gram precision
P4 = 4-gram precision

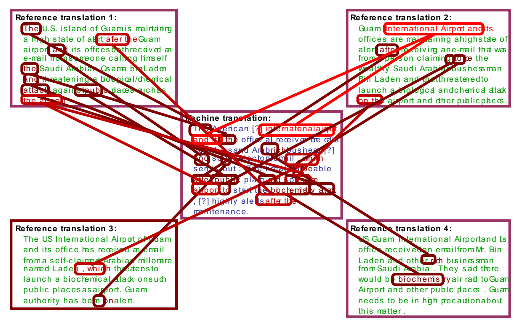
Note: only works at corpus level (zeroes kill it); there's a smoothed variant for sentence-level

BLEU in Action

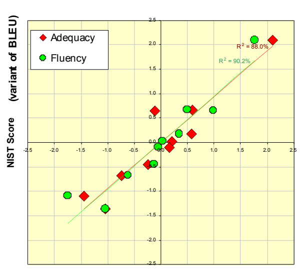
枪手被警方击毙。	**(Foreign Original)**
the gunman was shot to death by the police .	**(Reference Translation)**
the gunman was police kill .	#1
wounded police jaya of	#2
the gunman was shot dead by the police .	#3
the gunman arrested by police kill .	#4
the gunmen were killed .	#5
the gunman was shot to death by the police .	#6
gunmen were killed by police ?SUB>0 ?SUB>0	#7
al by the police .	#8
the ringer is killed by the police .	#9
police killed the gunman .	#10

green = 4-gram match (good!)
red = word not matched (bad!)

Multiple Reference Translations



Initial results showed that BLEU predicts human judgments well



slide from G. Doddington (NIST)

Automatic evaluation of MT

- People started optimizing their systems to maximize BLEU score
  - BLEU scores improved rapidly
  - The correlation between BLEU and human judgments of quality went way, way down
  - StatMT BLEU scores now approach those of human translations but their true quality remains far below human translations
- Coming up with automatic MT evaluations has become its own research field
  - There are many proposals: TER, METEOR, MaxSim, SEPIA, our own RTE-MT
  - TERpA is a representative good one that handles some word choice variation.
- MT research **requires** *some* automatic metric to allow a rapid development and evaluation cycle.

Pots of data

- You build a model on a **training set**.
- Commonly, you then set further hyperparameters on another set of data, the **tuning set**
  - But it's the training set for the hyperparameters
- You measure progress as you go on a **dev set** (development test set)
  - If you do that a lot you overfit to the dev set so it's good to have a second dev set, **dev2** set
- You evaluate and present final numbers on a **test set**

55

Pots of data

- For different reasons, the **train**, **tune**, **dev**, and **test** sets need to be completely distinct
- It is invalid to test on material you have trained on.
- If you keep running on the same evaluation set, you also begin to overfit to the evaluation set
  - Effectively you are "training" on the evaluation set … you are learning things that do and don't work on that particular training set.
- To get a valid measure of system performance you need another **independent test set**
  - Ideally, you only test on it once … definitely very few times

56