Tokens and Word Vectors: Subwords, BPE, and word2vec

Lecture Notes Consolidated with Claude AI — CS6140 Machine Learning

---

Instructor's lecture plan for the note of the same title (`TOKENS_claude_1017.pdf`). Section numbers (§) refer to that note.

Week 13, first lecture (Mon 11/30, the day before HW6 is due); it unlocks HW7 Problem 1. It assumes HW6 Problem 3 (word tokens, `nn.Embedding`), HW2's logistic regression and Week 5's PCA. This is the lightest lecture of Weeks 13–14 on purpose: the two ideas are counting (BPE) and logistic regression (word2vec), and the lecture after this one (attention) is the hardest of the term. Companion materials: the annotated lecture 8/13 (word vectors) and **TOKENS_ADVANCED** (optional).

HW7's sessions: Mon 11/30 (this plan), Thu 12/3 (ATTENTION Lecture A), Mon 12/7 (ATTENTION Lecture B + TRANSFORMER Part A), the extra lecture (TRANSFORMER Part B) and Thu 12/10 (a workshop: second pass at the decoder and GPT). If the extra lecture cannot be held between 12/7 and 12/10, Part B moves to 12/10 and the workshop is dropped. Mon 12/14 (HW7 is due the next day) is optional and advanced only; the final exam is Thu 12/17.

Lecture plan (100 minutes, mostly on the board)

Time	On the board	Worked example / demo
0–8	HW6's pipeline, text → pieces → IDs → vectors → model, with the two boxes of today circled; leave it up all lecture. Parts table: only "merging pairs" and "noise words" are new. (§1)	—
8–22	Words vs characters vs subwords: what the model *receives*. Run the note's five-text table live (Table 2); the moral, "no modern LM reads whole words"; the LLM tokenizer table. (§2, §3)	`verry hapy elefant`, the Spanish sentence, the code line: 3 tokenizers
22–45	BPE by hand: count pairs weighted by word frequency, merge, recount. Students call out the counts. Then encode unseen words by applying the merges in order. Ties: what to expect against the library. (§4)	The `hug/pug/pun` corpus: (u,g) 8, (h,ug) 6, (u,n) 4; `sung` → `s un g`
45–55	One-hot vs an embedding table; lookup = one-hot $\times E$; cosine vs dot product. (§5)	$(3,3)$ vs $(1,0)$, $(0,2)$: cosines 0.71, 0.71
55–70	word2vec as a fake task: redraw the 8/13 p. 5 network; skip-gram's softmax and why its denominator is too expensive. (§6)	"the black **cat** sat on", window 2
70–88	The p. 8 tricks board: a yes/no question per pair, noise words, and **SGNS = logistic regression**; the loss; subsampling; windows inside one story. (§7)	Scores 2, −1, 0.5 → 1.4143, with the three gradient signs
88–96	What the vectors learn: neighbours, related vs similar, the PCA picture; why two word2vec runs never give identical lists. (§8)	Demo (3 min): neighbours and the PCA plot from the reference run
96–100	HW7 Problem 1 map and the common bugs; forward pointer: attention turns these static vectors into contextual ones. (§9, §10)	—

Board layout suggestion. Keep the BPE pair-count table on one panel for the whole BPE segment (one row per step, struck-through counts as pairs disappear); students learn more from watching (h,u) 6 turn into (h,ug) 6 than from the final merge list. Use a second panel for the word2vec network and draw the logistic-regression reading of negative sampling directly on top of it.

The story, in five beats

1. **The model only ever sees vectors.** Everything before the first layer is a design choice: what counts as a piece, and which vector each piece gets. HW6 chose whole words and random vectors. Today replaces both choices.

2. **Words break the moment the text is not TinyStories.** Typos, names, other languages and code become `<unk>`, which the model can neither read nor write. Characters fix that but make sequences long. Subwords are the compromise every modern model uses.

3. **BPE is counting.** Glue the most frequent neighbouring pair, recount, repeat. The order of the merges is the tokenizer, and it splits any new word into known pieces. The only subtlety students will meet in the HW is a tie.

4. **A vector per piece that knows similarity, from a fake task.** Predicting neighbours forces words with similar contexts to similar vectors. The full softmax is too expensive, so ask a yes/no question per pair instead — and that question is HW2's logistic regression.

5. **Static now, contextual next.** One vector per word cannot tell *bank* from *bank*. Attention, next lecture, rewrites each vector using its neighbours in the sentence.