Language models

Language model

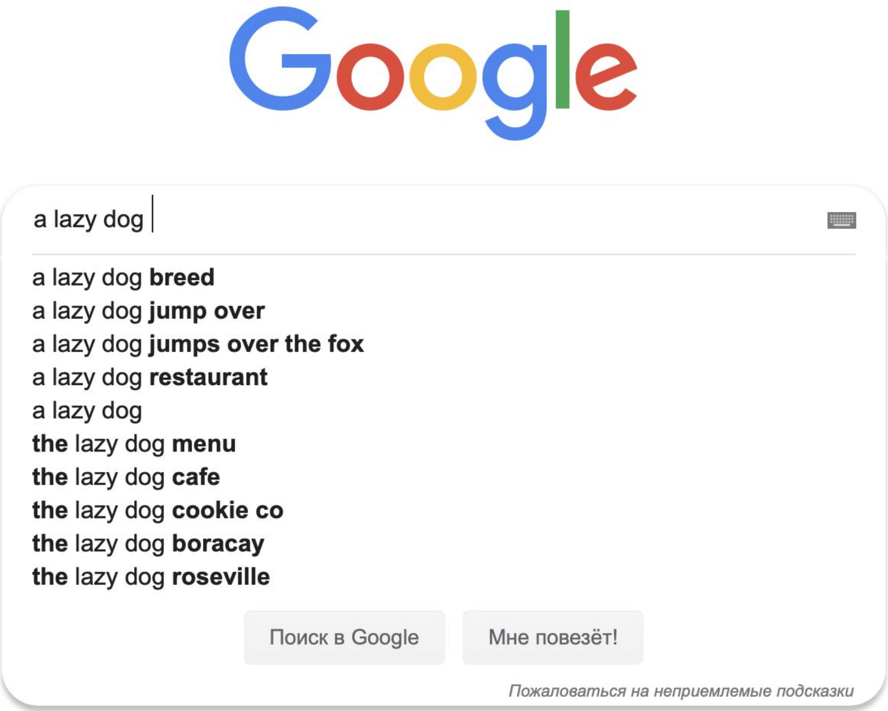
Language models estimate **probabilities** of various **tokens**:

- words
- letters
- …

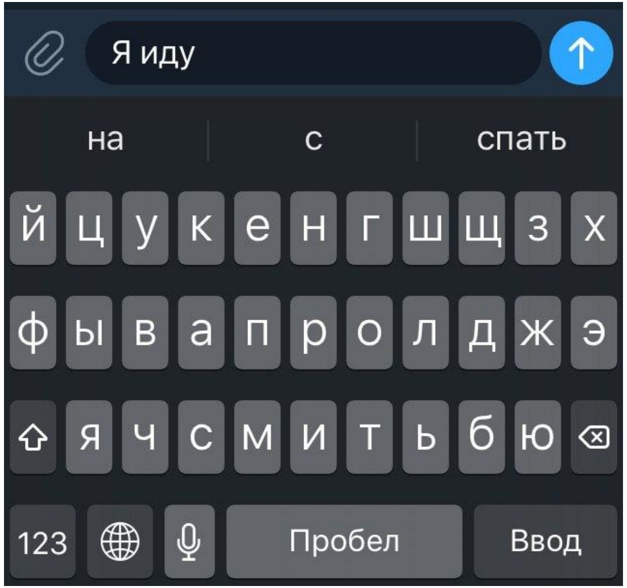
or estimate **probability** of **token sequences**:

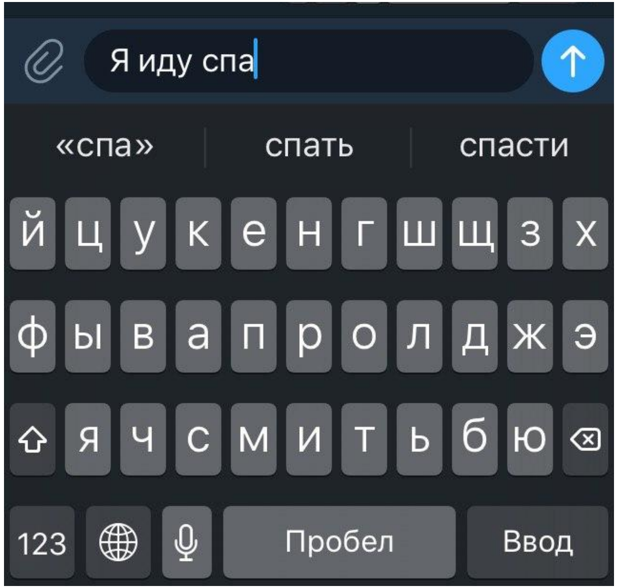
- sentences
- words
- ...

Language model

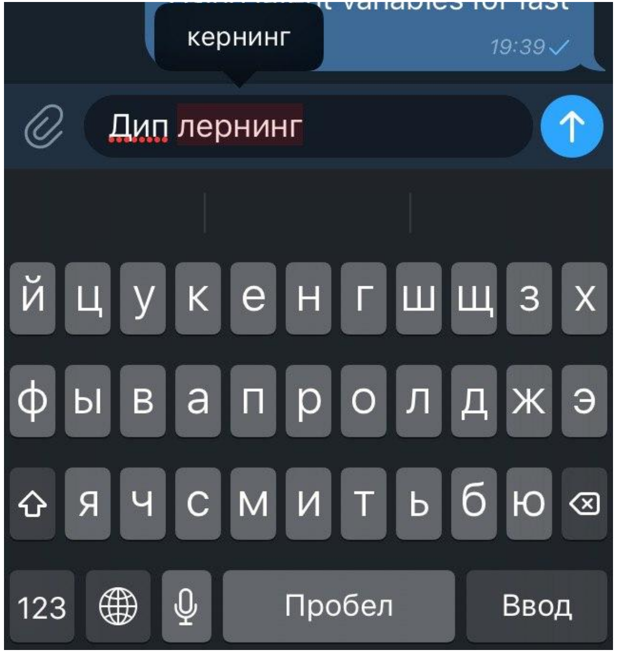


Language model





Language model



Language model

Token probability: how likely this token is to occur after a sequence of tokens

I am going to

- sleep
- study
- school

Sentence probability: how likely this sentence is to occur in natural language

P(I am going to sleep) > P(I am sleep going to)

Language models

Count-based

Neural

Count-based LM

Probability estimation

Let’s count different Bertie Bott’s tastes in ten sweets pack:

Banana	Black pepper	Cinnamon	Lemon	Dirt	Earthworm	Grass
26	23	19	20	17	13	22

$$P(taste) = \frac{Count(taste)}{\sum Count(taste)}$$

Probability estimation: sentences

- we want to estimate sentence probabilities using same idea
- we have large corpus of data
- we want to be able to estimate probability of a sentence that **we have never seen before**

The Ftaghn credled raftangly spattengly

Ftagn ranght kdufbfnms msnlkvrl

P = ?

Probability estimation: sentences

Problems:

- we always have small training data
- we have many sentences that occure in training set 0 times

Probability estimation: sentences

Problems:

- we always have small training data
- we have many sentences that occure in training set 0 times

Solution:

- estimate sentence probability as combination of probabilities of its smaller parts: **N-gram Language model**

Probability estimation: sentences

We want to get $P(\text{I want to sleep})$

Idea #1:

1. $P(\text{I want to sleep}) = P(w1 = \text{I}, w2 = \text{want}, w3 = \text{to}, w4 = \text{sleep})$

2. $P(X, Y) = P(Y| X)\, P(X)$

   $P(\text{I want to sleep}) = P(\text{sleep} \mid \text{I want to}) \cdot P(\text{I want to}) =$

   $= P(\text{sleep} \mid \text{I want to}) \cdot P(\text{to} \mid \text{I want}) \cdot P(\text{want} \mid \text{I}) \cdot P(\text{I})$

Probability estimation: sentences

$P(\text{I want to sleep}) = P(\text{sleep} \mid \text{I want to}) \cdot P(\text{I want to}) =$

$= P(\text{sleep} \mid \text{I want to}) \cdot P(\text{to} \mid \text{I want}) \cdot P(\text{want} \mid \text{I}) \cdot P(\text{I})$

Problem:

we still need to estimate probabilities of huge prefixes:

$P(\text{A cute little sheep was enjoying its life}) =$

$P(\text{life} \mid \text{A cute little sheep was enjoying its}) \cdot \ldots$

Probability estimation: sentences

Solution: **Independence assumption**:

next word depends only on **n** previous words:

- **trigram model:** $P(w_i \mid w_1, w_2, \ldots, w_{i-1}) \sim P(w_i \mid w_{i-1}, w_{i-2})$
- **bigram model:** $P(w_i \mid w_1, w_2, \ldots, w_{i-1}) \sim P(w_i \mid w_{i-1})$
- **unigram model**: $P(w_i \mid w_1, w_2, \ldots, w_{i-1}) \sim P(w_i)$

Probability estimation: sentences

Finally estimating probabilities:

$$P(w_i \mid w_1, \ldots, w_{i-1}) = P(w_i \mid w_{i-k}, \ldots, w_{i-1}) =$$

$$\frac{Count(w_i, w_{i-1}, \ldots, w_{i-k})}{Count(w_{i-1}, \ldots, w_{i-k})}$$

$$P(sleep \mid I\ am\ going\ to) = P(sleep \mid going\ to) = \frac{Count(going\ to\ sleep)}{Count(going\ to)}$$

N-gram LM

~~The fox runs~~ $\underbrace{\text{fast to the}}$ __

$$P(w_i \mid The\ fox\ runs\ fast\ to\ the) = P(w_i \mid fast\ to\ the) = \frac{Count(fast\ to\ the\ w_i)}{Count(fast\ to\ the)}$$

N-gram LM

Problems:

$$P(w_i \mid The\ fox\ runs\ fast\ to\ the) = P(w_i \mid fast\ to\ the) = \frac{Count(fast\ to\ the\ w_i)}{Count(fast\ to\ the)}$$

What if we didn't see this prefix
in training set at all?

N-gram LM

Problems:

$$P(w_i \mid The\ fox\ runs\ fast\ to\ the) = P(w_i \mid fast\ to\ the) = \frac{Count(fast\ to\ the\ w_i)}{Count(\boxed{fast\ to\ the})}$$

What if we didn't see this prefix
in training set at all?

Solution: **backoff**:

$$P(w_i \mid fast\ to\ the) = \frac{Count(fast\ to\ the\ w_i)}{Count(\boxed{to\ the})}$$

Language models

How to score language models:

1. Cross-entropy / perplexity

   score sentences from validation set using our LM:

$$H_M(w_1, \dots w_n) = -\frac{1}{n} P_M(w_1, \dots w_n)$$

2. Put LM into another NLP system and see if this works better

   (this is too complicated)

Bigram language model

The __

Bigram language model

The __

cat	0.35
dog	0.3
man	0.15
fox	0.05
clock	0.0006
onion	0.00004
go	0.00000000012

Bigram language model

The cat

Bigram language model

The cat __

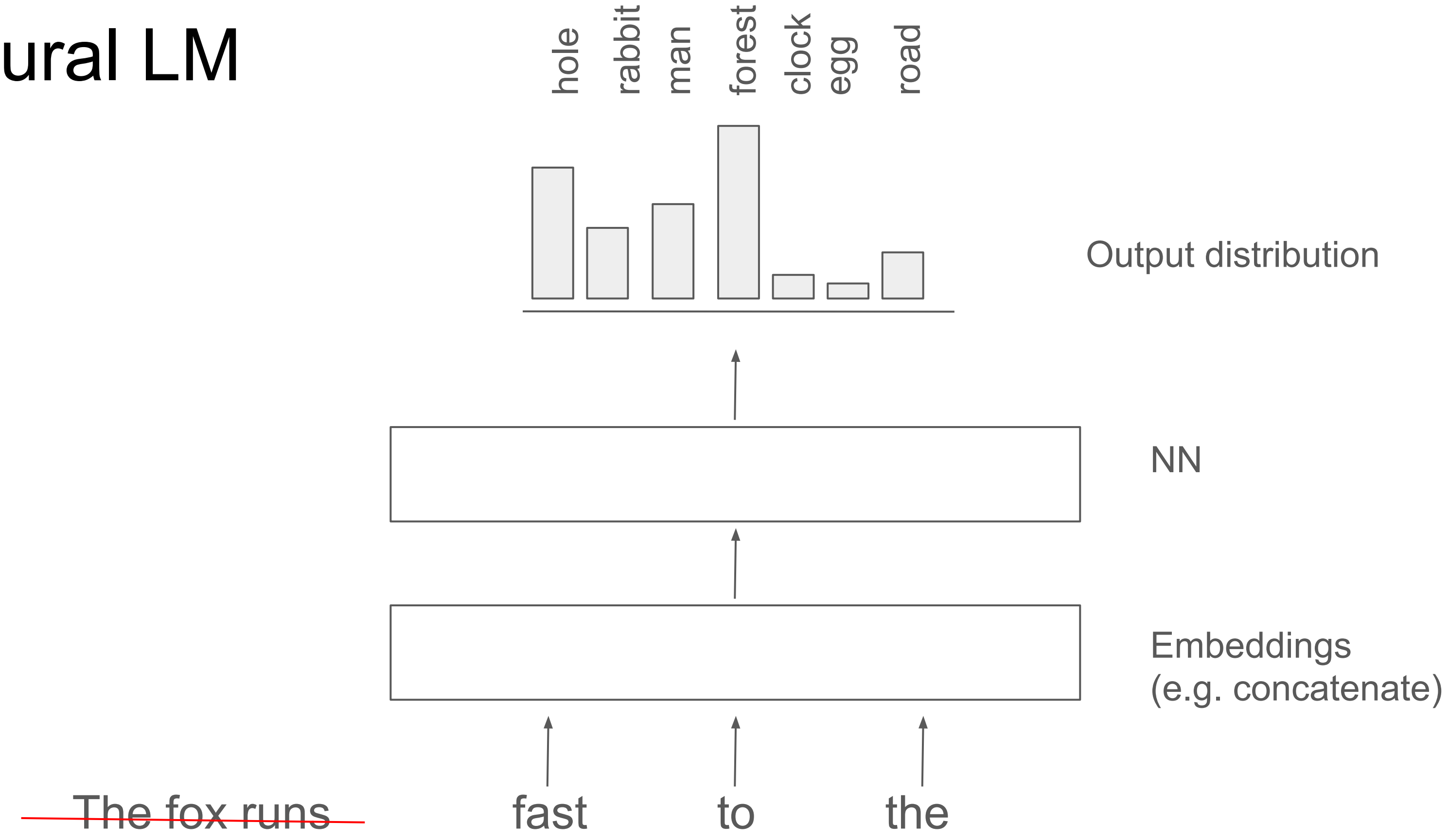
lays	0.44
sits	0.12
runs	0.06
run	0.0002
stays	0.0006
onion	0.000009
go	0.000000000012

Bigram language model

The cat lays __

Neural LM

Neural LM


hole
rabbit
man
forest
clock
egg
road
Output distribution
NN
Embeddings
(e.g. concatenate)
~~The fox runs~~
fast
to
the

Neural LM

Advantages:

- no problem with unseen words
- no problem with zero probabilities

Neural LM

Advantages:

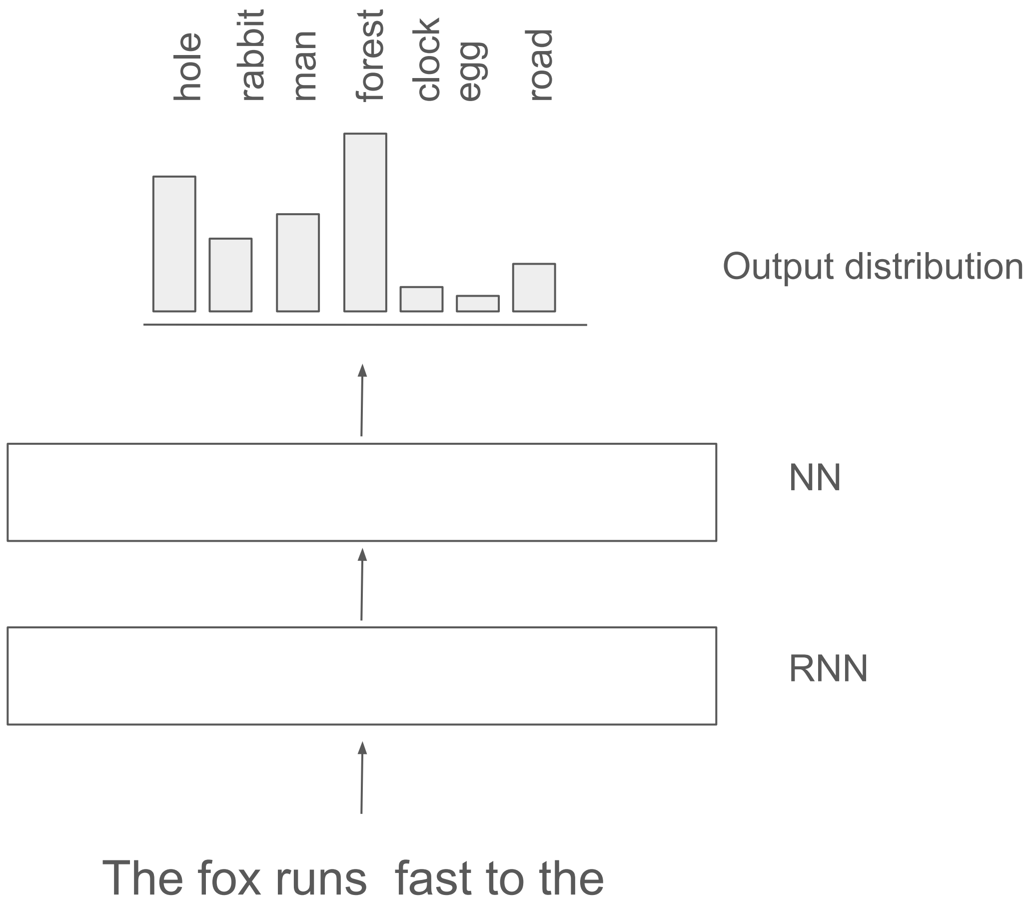
- no problem with unseen words
- no problem with zero probabilities

Remaining problems:

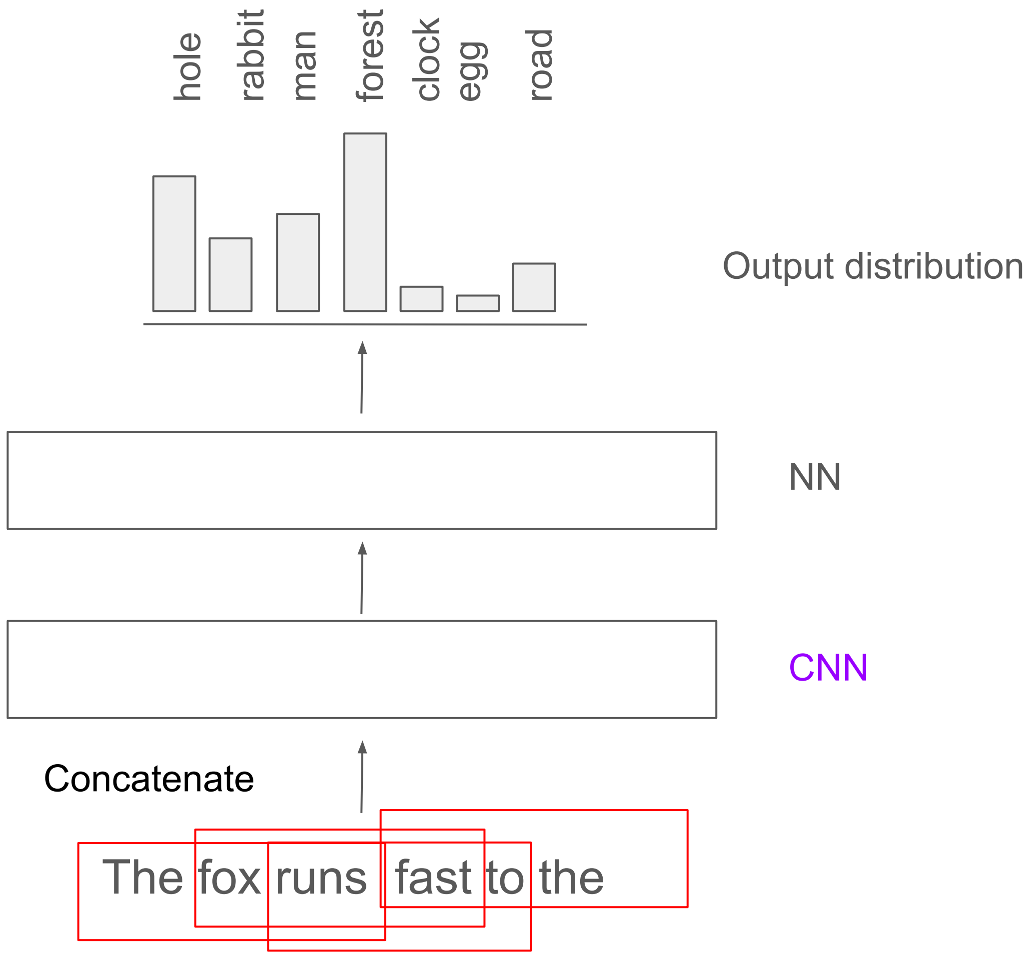
- markov property
- word processing depends on its position

The fox runs fast to the

Neural LM


hole
rabbit
man
forest
clock
egg
road
Output distribution
NN
RNN
The fox runs fast to the

Neural LM


hole
rabbit
man
forest
clock
egg
road
Output distribution
NN
CNN
Concatenate
The fox runs fast to the

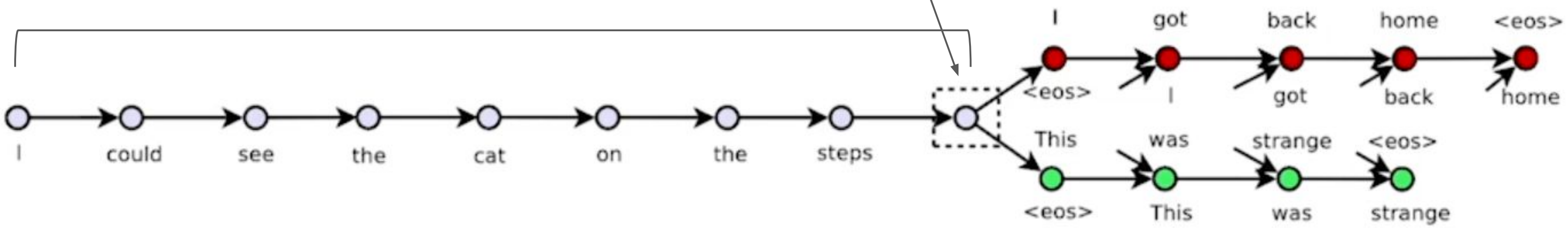
LM for Embeddings

Why not to get embeddings from language model?

Sentence embeddings

Generate next sentence
word-by-word

RNN



LM examples

For $\bigoplus_{n=1,\ldots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, **??** and the fact that any U affine, see Morphisms, Lemma **??**. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma **??** we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example **??**. It may replace S by $X_{spaces,étale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma **??**. Namely, by Lemma **??** we see that R is geometrically regular over S.

Lemma 0.1. *Assume (3) and (3) by the construction in the description.*

Suppose $X = \lim |X|$ (by the formal open covering X and a single map $\underline{Proj}_X(\mathcal{A}) = \mathrm{Spec}(B)$ over U compatible with the complex

$$Set(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X,\mathcal{O}_X}).$$

*When in this case of to show that $\mathcal{Q} \to \mathcal{C}_{Z/X}$ is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition **??** (without element is when the closed subschemes are catenary. If T is surjective we may assume that T is connected with residue fields of S. Moreover there exists a closed subspace $Z \subset X$ of X where U in X' is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem*

(1) *f is locally of finite type. Since $S = \mathrm{Spec}(R)$ and $Y = \mathrm{Spec}(R)$.*

Proof. This is form all sheaves of sheaves on X. But given a scheme U and a surjective étale morphism $U \to X$. Let $U \cap U = \coprod_{i=1,\ldots,n} U_i$ be the scheme X over S at the schemes $X_i \to X$ and $U = \lim_i X_i$. □

The following lemma surjective restrocomposes of this implies that $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{\mathcal{X},\ldots,0}$.

Lemma 0.2. *Let X be a locally Noetherian scheme over S, $E = \mathcal{F}_{X/S}$. Set $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$. Since $\mathcal{I}^n \subset \mathcal{I}^n$ are nonzero over $i_0 \leq \mathfrak{p}$ is a subset of $\mathcal{J}_{n,0} \circ \overline{A}_2$ works.*

Lemma 0.3. *In Situation **??**. Hence we may assume $\mathfrak{q}' = 0$.*

Proof. We will use the property we see that $\mathfrak{p}$ is the mext functor (**??**). On the other hand, by Lemma **??** we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where K is an F-algebra where δ_{n+1} is a scheme over S. □

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

LM examples

```
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
  rw->name = "Getjbbregs";
  bprm_self_clearl(&iv->version);
  regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECONDS << 12;
  return segtable;
}
```

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

The sole importance of the crossing of the Berezina lies in the fact that it plainly and indubitably proved the fallacy of all the plans for cutting off the enemy's retreat and the soundness of the only possible line of action--the one Kutuzov and the general mass of the army demanded--namely, simply to follow the enemy up. The French crowd fled at a continually increasing speed and all its energy was directed to reaching its goal. It fled like a wounded animal and it was impossible to block its path. This was shown not so much by the arrangements it made for crossing as by what took place at the bridges. When the bridges broke down, unarmed soldiers, people from Moscow and women with children who were with the French transport, all--carried on by vis inertiae--pressed forward into boats and into the ice-covered water and did not, surrender.

https://arxiv.org/pdf/1506.02078.pdf

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
    siginfo_t *info)
{
  int sig = next_signal(pending, mask);
  if (sig) {
    if (current->notifier) {
      if (sigismember(current->notifier_mask, sig)) {
        if (!(current->notifier)(current->notifier_data)) {
          clear_thread_flag(TIF_SIGPENDING);
          return 0;
        }
      }
    }
    collect_signal(sig, pending, info);
  }
  return sig;
}
```

```
#ifdef CONFIG_AUDITSYSCALL
static inline int audit_match_class_bits(int class, u32 *mask)
{
  int i;
  if (classes[class]) {
    for (i = 0; i < AUDIT_BITMASK_SIZE; i++)
      if (mask[i] & classes[class][i])
        return 0;
  }
  return 1;
}
```

https://arxiv.org/pdf/1506.02078.pdf