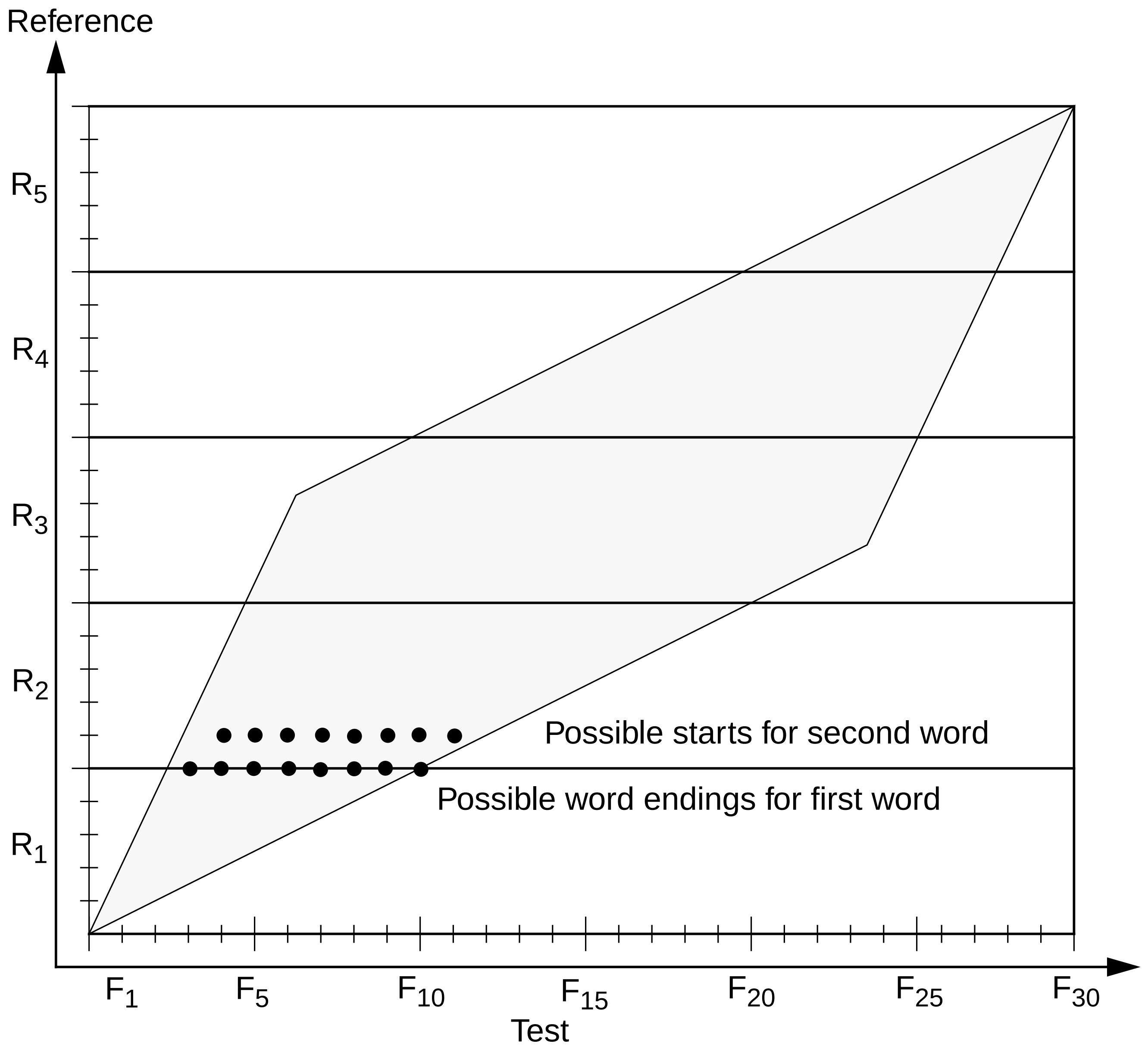
DTW, Syntactic Constraints, and Beam Search

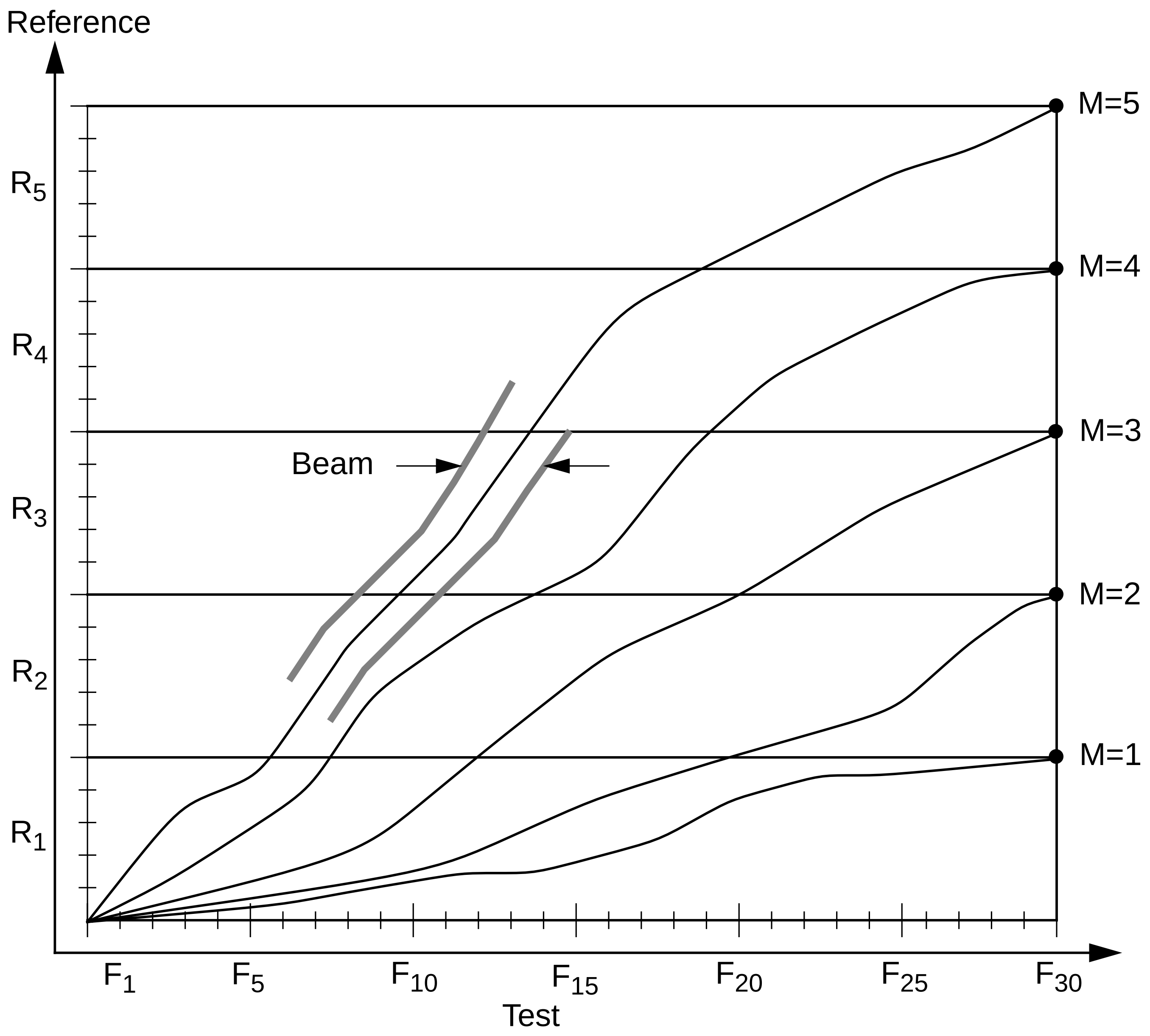
Consider the problem of connected digit recognition: “325 1739”. In the simplest case, any digit can follow any other digit, but we might know the exact number of digits spoken.

An elegant solution to the problem of finding the best overall sentence hypothesis is known as level building (typically assumes models are same length.


Reference
R5
R4
R3
R2
R1
Possible starts for second word
Possible word endings for first word
F1
F5
F10
F15
F20
F25
F30
Test

• Though this algorithm is no longer widely used, it gives us a glimpse into the complexity of the syntactic pattern recognition problem.

Level Building For An Unknown Number Of Words

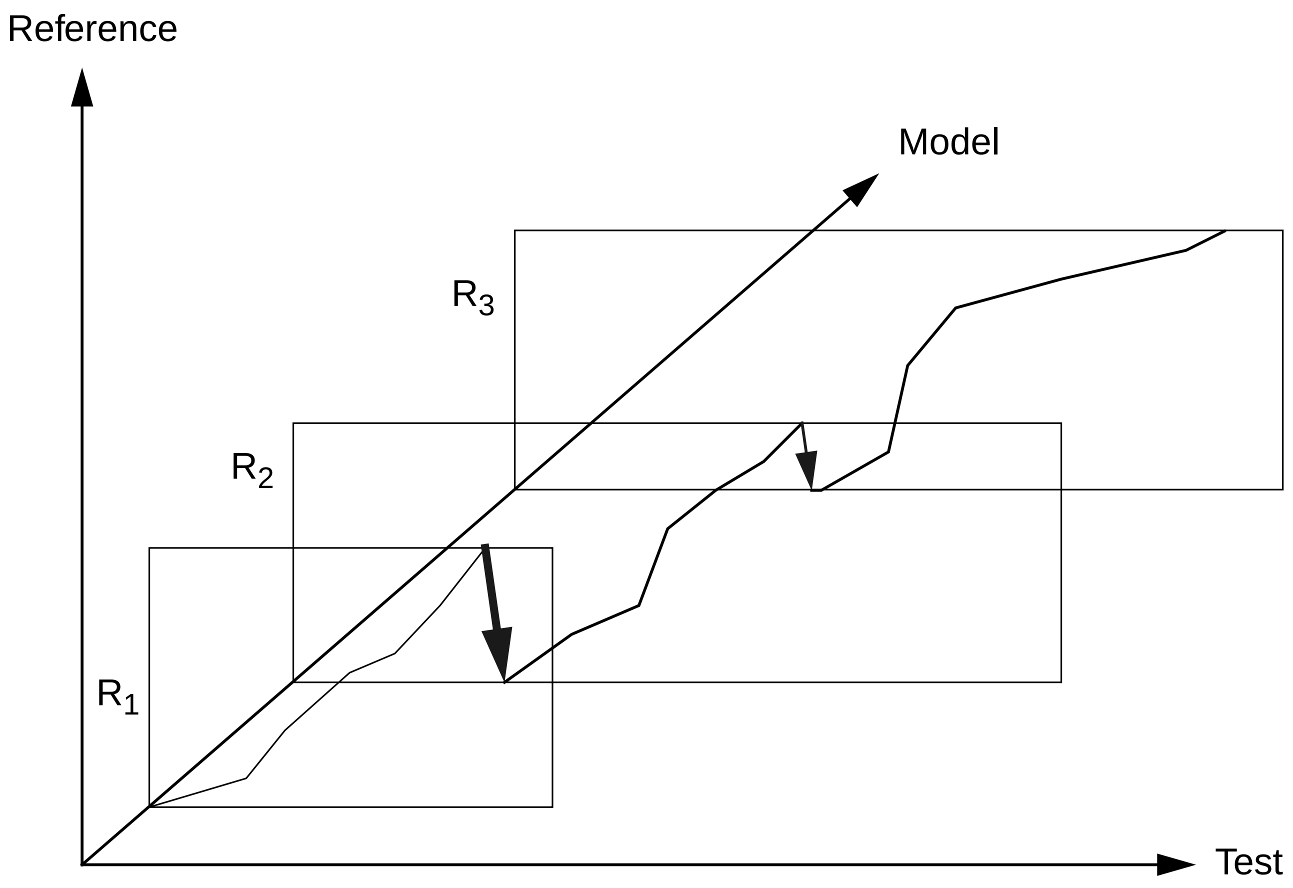


- Paths can terminate on any level boundary indicating a different number of words was recognized (note the significant increase in complexity)
- A search band around the optimal path can be maintained to reduce the search space
- Next-best hypothesis can be generated (N-best)
- Heuristics can be applied to deal with free endpoints, insertion of silence between words, etc.
- Major weakness is the assumption that all models are the same length!

The One-Stage Algorithm (“Bridle Algorithm”)

The level building approach is not conducive to models of different lengths, and does not make it easy to include syntactic constraints (which words can follow previous hypothesized words).

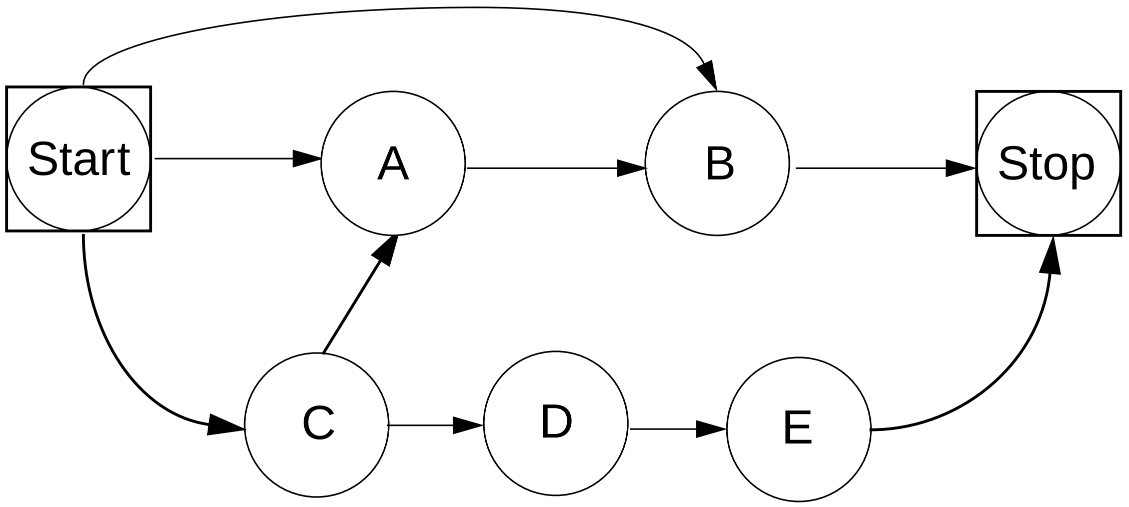
An elegant algorithm to perform this search in one pass is demonstrated below:



- Very close to current state-of-the-art doubly-stochastic algorithms (HMM)
- Conceptually simple, but difficult to implement because we must remember information about the interconnections of hypotheses
- Amenable to beam-search concepts and fast-match concepts
- Supports syntactic constraints by limited the choices for extending a hypothesis
- Becomes complex when extended to allow arbitrary amounts of silence between words
- How do we train?

Introduction of Syntactic Information

- The search space for vocabularies of hundreds of words can become unmanageable if we allow any word to follow any other word (often called the no-grammar case)
- Our rudimentary knowledge of language tells us that, in reality, only a small subset of the vocabulary can follow a given word hypothesis, but that this subset is sensitive to the given word (we often refer to this as “context-sensitive”)
- In real applications, user-interface design is crucial (much like the problem of designing GUI’s), and normally results in a specification of a language or collection of sentence patterns that are permissible
- A simple way to express and manipulate this information in a dynamic programming framework is a via a state machine:



For example, when you enter state C, you output one of the following words: {daddy, mommy}.

If:

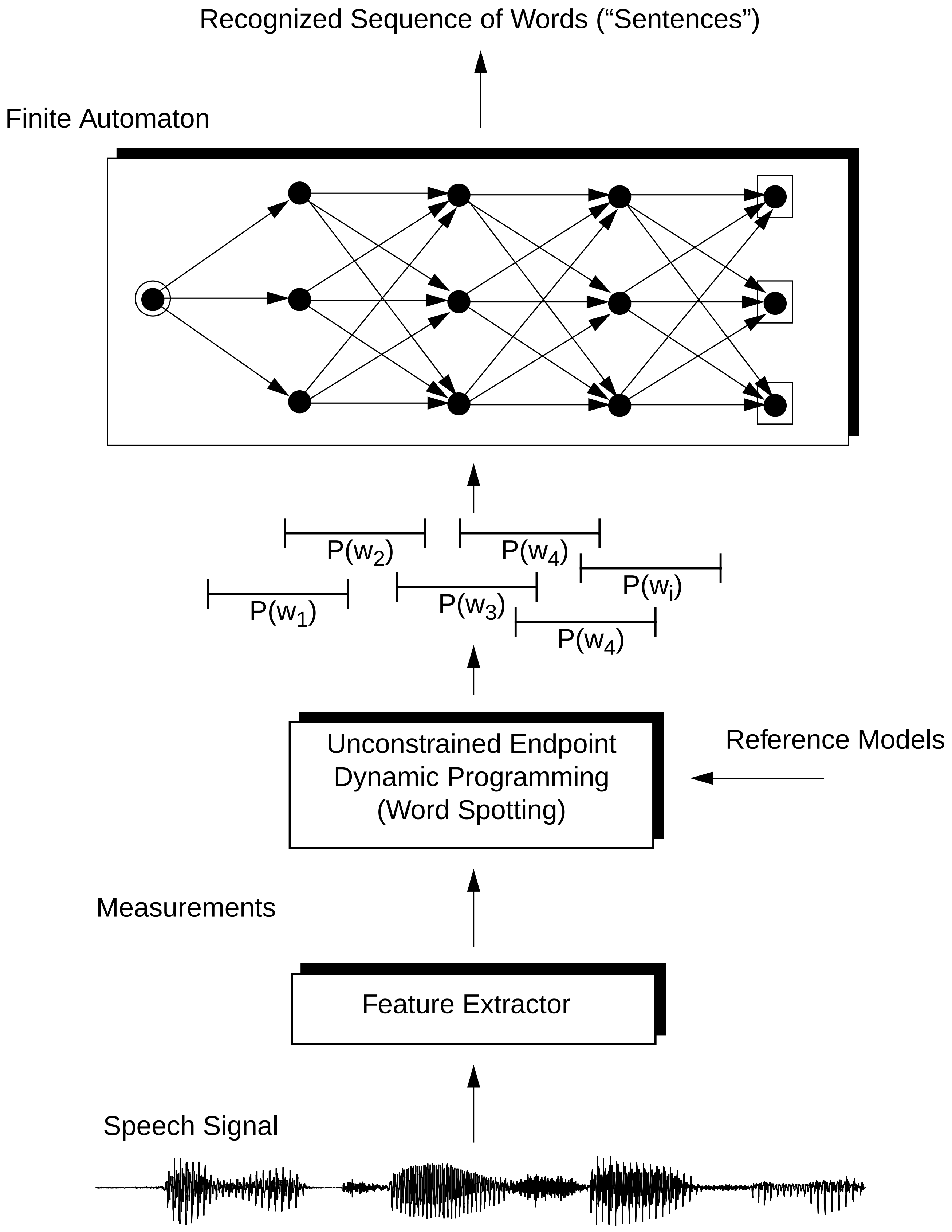
state A:	give
state B:	me
state C:	{daddy, mommy}
state D:	come
state E:	here

We can generate phrases such as:

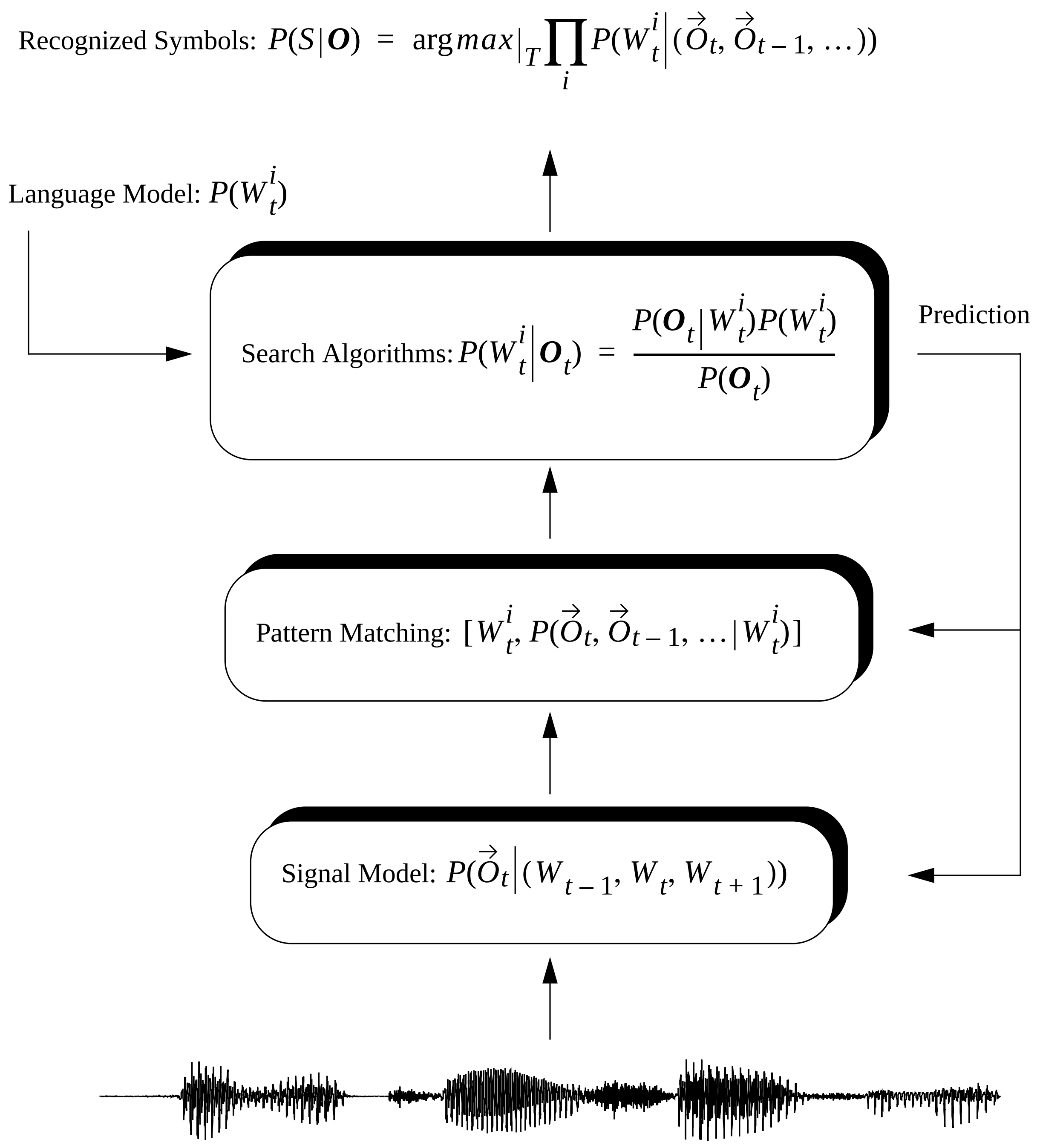
Daddy give me

- We can represent such information numerous ways (as we shall see)

Early Attempts At Introducing Syntactic Information Were “Ad-Hoc”

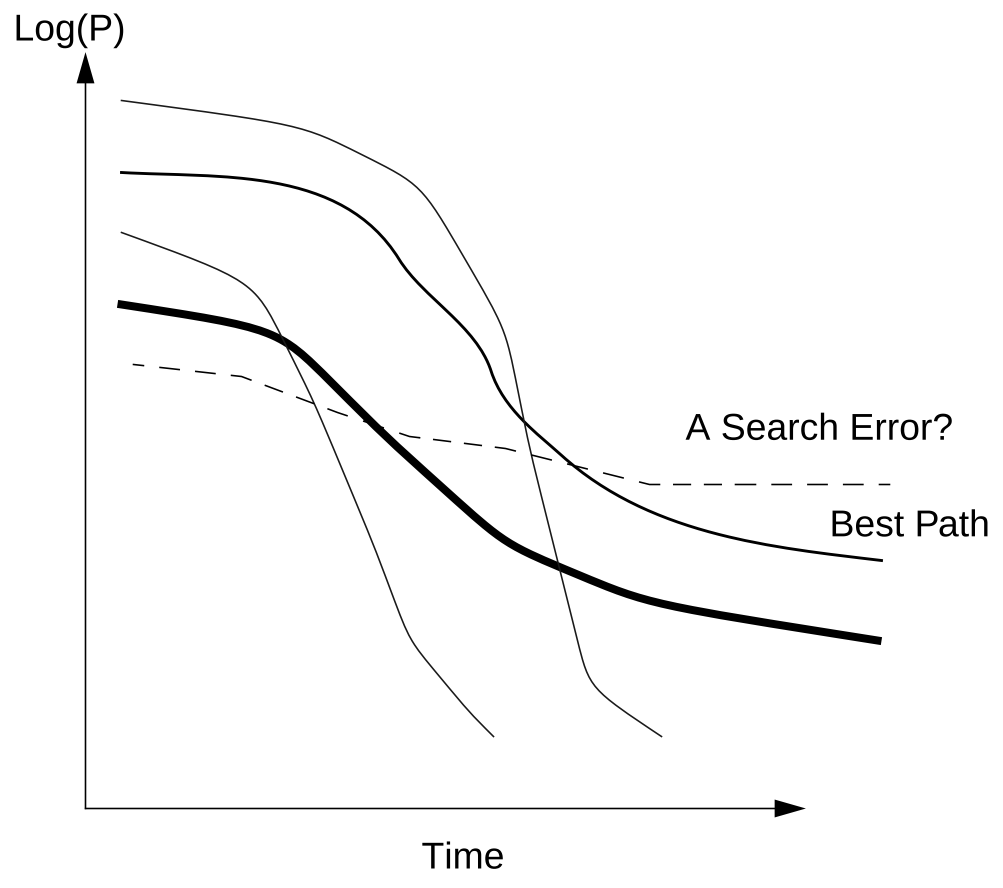


BASIC TECHNOLOGY — A PATTERN RECOGNITION PARADIGM BASED ON HIDDEN MARKOV MODELS


Recognized Symbols: $P(S|\boldsymbol{O}) = \arg max|_T \prod_i P(W_t^i | (\vec{O}_t, \vec{O}_{t-1}, \ldots))$
Language Model: $P(W_t^i)$
Search Algorithms: $P(W_t^i | \boldsymbol{O}_t) = \frac{P(\boldsymbol{O}_t | W_t^i) P(W_t^i)}{P(\boldsymbol{O}_t)}$
Prediction
Pattern Matching: $[W_t^i, P(\vec{O}_t, \vec{O}_{t-1}, \ldots | W_t^i)]$
Signal Model: $P(\vec{O}_t | (W_{t-1}, W_t, W_{t+1}))$

BEAM SEARCH

- The modern view of speech recognition is a problem consisting of two operations: signal modeling and search.
- Finding the most probable sequence of words is an optimization problem that can be solved by dynamic programming
- Optimal solutions are intractable; fortunately, sub-optimal solutions yield good performance
- Beam search algorithms are used to trade-off complexity vs. accuracy


Log(P)
A Search Error?
Best Path
Time